Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

State-dependent model predictive control of nonlinear vehicle dynamics with implicit state-derivative coupling and actuator dynamics

Javad Zolfaghari

gs64011@student.upm.edu.my

Universiti Putra Malaysia

Ahmad Salahuddin Mohd Harithuddin

Universiti Putra Malaysia

Nuraini Abdul Aziz

Universiti Putra Malaysia

Raja Mohd Kamil Raja Ahmad

Universiti Putra Malaysia

Reza N Jazar

Tarleton State University

Research Article

Keywords:

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11471168/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

State-dependent model predictive control of nonlinear vehicle dynamics with implicit state-derivative coupling and actuator dynamics

Javad Zolfaghari[1*], Ahmad Salahuddin Mohd Harithuddin[1], Nuraini Abdul Aziz[2], Raja Mohd Kamil Raja Ahmad[3], Reza N Jazar[4]

[1*]Department of Aerospace Engineering, Universiti Putra Malaysia, Serdang, Selangor, Malaysia.
[2]Department of Mechanical and Manufacturing Engineering, Universiti Putra Malaysia, Serdang, Selangor, Malaysia.
[3]Department of Electrical and Electronic Engineering, Universiti Putra Malaysia, Serdang, Selangor, Malaysia.
[4]Department of Mechanical, Environmental and Civil Engineering, Tarleton State University, Texas, USA.

*Corresponding author(s). E-mail(s): gs64011@student.upm.edu.my;

Abstract

Longitudinal load transfer turns the equations of motion of a rear-wheel-drive vehicle at the handling limit into an implicit ordinary differential equation, in which the state derivative enters the tyre forces through the normal loads. This study seeks a controller that remains reliable for such systems while balancing accuracy and computing time. The implicit equation is resolved by augmenting the state with the accelerations and the actuator states. This gives a state-dependent coefficient (SDC) model that carries the steady-state condition in its structure, so the equilibrium needs no trim input. On this model a state-dependent model predictive controller (SDMPC) is proposed. At each sample it builds its prediction model from SDC matrices evaluated along a rolled-out nonlinear trajectory and solves a single convex quadratic programme. It is compared with a state-dependent Riccati equation controller, linear MPC, linear time-varying MPC and nonlinear MPC, using initial conditions ordered by a two-criterion severity score. The Riccati-based and linear controllers fail at moderate severity. Linear time-varying MPC is cheaper on a severe condition but less

accurate, and it diverges on the most severe one, an opposite-direction near-spin state, where nonlinear MPC settles at an offset steady state at a much higher computing cost. SDMPC is the only controller that stabilises every condition tested, at a cost close to one sampling period in a MATLAB prototype. The conclusions apply to the vehicle model, test set and implementations studied.

Keywords: state-dependent coefficient, implicit ODE, model predictive control, vehicle drifting, nonlinear MPC

1 Introduction

Some nonlinear systems have a vector field that depends on the state derivative itself, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{u})$. Such *implicit ordinary differential equations* (implicit ODEs) have more structure than a general differential–algebraic equation (DAE) [1], yet they do not fit the explicit-form analyses on which Lyapunov stability, phase-plane characterisation and state-dependent Riccati methods are usually built.

Vehicle dynamics at the handling limit is a practical example. Longitudinal load transfer makes the vertical tyre loads F_{zf} and F_{zr} depend on $\dot{v}_x$, and the tyre law feeds these loads back into the acceleration that produces them. Sustained drifting shows the consequence clearly. It involves large body sideslip ($|\beta| > 20°$), rear-axle saturation under combined slip and open-loop instability [2, 5], and it is relevant to emergency avoidance, low-friction recovery and autonomous racing [4, 6]. Most studies use a planar bicycle model with states (v_x, v_y, r), find the drift equilibria from force and moment balance, linearise about them, and stabilise them with LQR, MPC or SDRE [3, 4, 7].

Two limitations are common to this body of work. First, vertical loads are usually taken as constant or quasi-static, which removes the dependence of F_{zf} and F_{zr} on $\dot{v}_x$ [10]. Retaining it gives the implicit ODE above, which a standard explicit integrator cannot advance until the algebraic coupling is resolved. Second, fixed-point linearisation confines the controller to a small neighbourhood of one operating point, although phase-plane studies reveal a rich global structure with multiple equilibria and separatrices [2, 9]. Nonlinear MPC removes the linearisation error [8], but it requires a non-convex programme at every sample, which remains the main obstacle to real-time use. Real-time-iteration and SQP schemes reduce this to one warm-started quadratic programme per sample [17, 18].

The question addressed here is practical: *which controller gives a reliable closed loop for an implicit-ODE system at an acceptable computational cost?* Linearisation-based schemes (LQR, LMPC, LTV-MPC) are cheap but lose accuracy away from the expansion point, while full nonlinear MPC is accurate in its model but expensive. We resolve the implicit equation by *augmenting* the state with the state derivatives and the actuator states, which gives the SDC form $\dot{\boldsymbol{z}} = \boldsymbol{A}(\boldsymbol{z})\,\boldsymbol{z} + \boldsymbol{B}(\boldsymbol{z})\,\boldsymbol{U}$. Because the derivatives are states, the steady-state condition is contained in the equations, the equilibrium needs no trim input, and consistency of the accelerations with the implicit equation is an invariant of the model (section 3.1). On this representation the

State-Dependent MPC (SDMPC) builds, at every sample, a prediction model from SDC matrices evaluated along a rolled-out nonlinear trajectory and solves a convex quadratic programme (QP).

The contributions are as follows. (i) The SDC resolution of the implicit vehicle ODE and its embedded steady-state structure. The consistency manifold is invariant with codimension three, and the frozen pair is structurally rank deficient, with three uncontrollable zero directions given by the gradients of the conserved consistency residuals (proofs in the appendices). (ii) The SDMPC formulation, with trajectory-wise SDC prediction, a matrix-exponential state transition and a fixed workload per step. (iii) A two-criterion severity ordering of the initial conditions, based on excess sideslip and phase-plane recovery distance. (iv) A controlled comparison of five controllers (SDRE, LMPC, LTV-MPC, NMPC with CasADi/IPOPT and SDMPC) on one plant, with all settings reported. The claims are limited to the model, test set and implementations of section 5.1.

2 Vehicle modelling and implicit dynamics

2.1 Equations of motion

The vehicle is a planar single-track model (figure 1). The inputs are the rear longitudinal slip ratio s_r, the front steering angle δ_f (positive counter-clockwise) and the front slip ratio s_f. With states $x_1 = v_x$, $x_2 = v_y$, $x_3 = r$, the equations of motion are [10]

$$\dot{x}_1 = \tfrac{1}{m}\textstyle\sum F_x + x_2 x_3, \quad \dot{x}_2 = \tfrac{1}{m}\textstyle\sum F_y - x_1 x_3, \quad \dot{x}_3 = \tfrac{1}{I_z}\textstyle\sum M_z, \tag{1}$$

with

$$\begin{aligned} \textstyle\sum F_x &= F_{xf}\cos\delta_f + F_{xr} - F_{yf}\sin\delta_f - C_A x_1^2, \\ \textstyle\sum F_y &= F_{yf}\cos\delta_f + F_{yr} + F_{xf}\sin\delta_f, \\ \textstyle\sum M_z &= a_1 F_{yf}\cos\delta_f + a_1 F_{xf}\sin\delta_f - a_2 F_{yr}. \end{aligned} \tag{2}$$

Longitudinal acceleration is coupled to the normal loads through

$$F_{zf} = mg\frac{a_2}{a_1+a_2} - m\frac{h}{a_1+a_2}(\dot{x}_1 - x_2 x_3), \qquad F_{zr} = mg\frac{a_1}{a_1+a_2} + m\frac{h}{a_1+a_2}(\dot{x}_1 - x_2 x_3). \tag{3}$$

The vehicle parameters are given in table 1.

2.2 Tyre forces: Modified Elliptical Method

The tyre forces are computed with the Modified Elliptical Method (MEM) [10, 12], which represents combined-slip coupling through two multiplicative saturation functions, ϕ and ϕ', that scale the pure-slip responses. The elliptical friction limit is thus contained in a closed-form, continuous expression suited to Jacobian-based analysis. With the forces normalised by the vertical loads,

$$F_{x\bullet} = \mathbf{MEM}_x(s_\bullet, \alpha_\bullet)\, F_{z\bullet}, \quad F_{y\bullet} = \mathbf{MEM}_y(s_\bullet, \alpha_\bullet)\, F_{z\bullet}, \quad \bullet \in \{f, r\}, \tag{4}$$

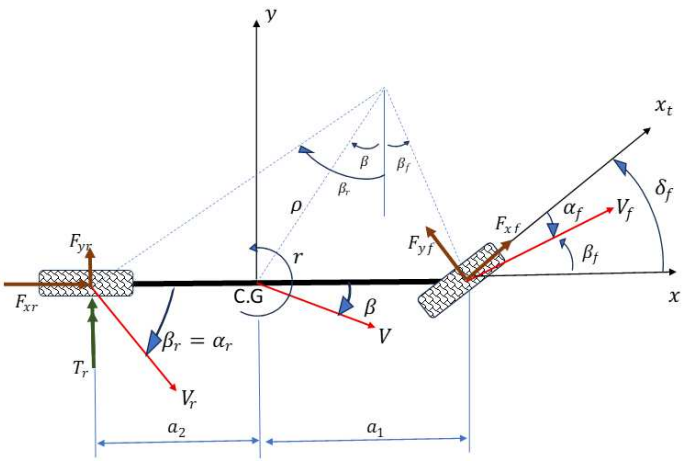


Fig. 1 Single-track model of the rear-wheel-drive vehicle. The front axle is at distance a_1 from the centre of gravity (c.g.) and the rear axle at a_2; δ_f and r are positive anticlockwise. V is the c.g. velocity, v_x and v_y its components, β the body sideslip angle and ρ the path radius.

Table 1 Vehicle parameters.

Parameter	Symbol	Value	Unit
Mass	m	800	kg
Yaw inertia	I_z	1000	kg·m^2
Distance to front axle	a_1	1.35	m
Distance to rear axle	a_2	1.50	m
c.g. height	h	0.90	m
Effective rolling radius	R_g	0.35	m
Aerodynamic drag coefficient	C_A	0.20	N·s^2/m^2

$$\mathbf{MEM}_x(s,\alpha) = \psi(s)\,\phi(\alpha), \qquad \mathbf{MEM}_y(s,\alpha) = -\psi'(\alpha)\,\phi'(s), \tag{5}$$

where $\psi(s)$ and $\psi'(\alpha)$ are the normalised pure-longitudinal and pure-lateral characteristics. The contact-patch velocities, slip ratios and slip angles are

$$v_{xtf} = v_x\cos\delta_f + (v_y + a_1 r)\sin\delta_f, \qquad v_{xtr} = v_x, \tag{6}$$

$$s_f = \frac{R\omega_f - v_{xtf}}{\max(R\omega_f,\, v_{xtf})}, \qquad s_r = \frac{R\omega_r - v_{xtr}}{\max(R\omega_r,\, v_{xtr})}, \tag{7}$$

$$\alpha_f = \arctan\!\left(\frac{v_y + a_1 r}{v_x}\right) - \delta_f, \qquad \alpha_r = \arctan\!\left(\frac{v_y - a_2 r}{v_x}\right), \tag{8}$$

with $R \equiv R_g$. The pure-slip and saturation functions are

$$\psi_s(s) = \operatorname{sgn}(s)\big[C_s s K(s, s_s, r_3, r_4)H(s, s_s) + (C_s s - b_1)J(s, s_s, r_1, r_2)(1 - H(s, s_s))\big], \tag{9}$$

$$\psi'_\alpha(\alpha) = \mathrm{sgn}(\alpha)\big[C_\alpha \alpha K(\alpha, \alpha_s, r'_3, r'_4) H(\alpha, \alpha_s) + (C_\alpha \alpha - b_2) J(\alpha, \alpha_s, r'_1, r'_2)(1 - H(\alpha, \alpha_s))\big], \tag{10}$$

$$\phi(\alpha) = \left[1 - D_{s\alpha} S(\alpha - \alpha_s)^2\right]^{1/n} J'(\alpha, \alpha_s, h_1, h_2), \qquad \phi'(s) = \left[1 - D_{\alpha s} S(s - s_s)^2\right]^{1/n'} J'(s, s_s, h'_1, h'_2), \tag{11}$$

with the auxiliary functions

$$\begin{aligned} J(p,q,u,v) &= \exp\left(-\frac{|p-q|^u}{v}\right), \qquad K(p,q,u,v) = 1 - \exp\left(-u\frac{|p-q|^v}{q}\right), \\ J'(p,q,u,v) &= \begin{cases} -q, & p < -q \\ p, & |p| \le q \\ q, & p > q \end{cases}, \qquad S(p,q) = \begin{cases} 0, & |p| < q \\ p, & |p| \le q \\ q, & p > q \end{cases}, \\ H(a,b) &= \begin{cases} 0, & a < b \\ 1, & a \ge b \end{cases}. \end{aligned} \tag{12}$$

K and J describe the post-saturation gain build-up and exponential decay, H separates the pre- and post-saturation regimes, J' clips the forces to the physical peak range and S models the loss of stiffness under combined slip, which reproduces the peak-and-decay shape of Magic Formula curves [11]. The discontinuous functions are replaced by smooth approximations, $\mathrm{sgn}(p) \approx \tanh(kp)$, $H(p,q) \approx \frac{1}{2}[1 + \tanh(k(q - |p|))]$ and $S(x,\ell) = xH(x,\ell) + \mathrm{sgn}(x)\ell[1 - H(x,\ell)]$, with $k \gg 1$. The MEM parameters, fitted to Magic Formula data [11, 12], are listed in table 2.

Table 2 Tyre parameters used in the MEM.

$C_{sf} = 8.6845$	$C_{sr} = 8.6845$	$C_{\alpha f} = 5.49$	$C_{\alpha r} = 5.49$
$s_s = 0.12$	$\alpha_s = 0.16$	$a = 0.6181$	$b = 0.6653$
$b_1 = 0.64$	$b_2 = 0.7727$	$n = -1.95$	$n' = -1.6$
$r_1 = 0.8$	$r_2 = 0.7$	$r_3 = 2.1$	$r_4 = 1.15$
$r'_1 = 1.25$	$r'_2 = 0.25$	$r'_3 = 2.5$	$r'_4 = 1.1$
$h_1 = 0.8$	$h_2 = 0.42$	$h'_1 = 0.86$	$h'_2 = 0.35$

2.3 Implicit ODE structure

Substituting (2)–(12) into (1) gives

$$\dot{x}_i = f_i(x_1, x_2, x_3, \dot{x}_1, s_f, s_r, \delta_f), \quad i = 1, 2, 3, \tag{13}$$

where, with $\mathbf{MEM}^f_{x,y} = \mathbf{MEM}_{x,y}(s_f, \alpha_f)$ and $\mathbf{MEM}^r_{x,y} = \mathbf{MEM}_{x,y}(s_r, \alpha_r)$,

$$
\begin{aligned}
f_1 &= \frac{\mathbf{MEM}^f_x F_{zf} \cos\delta_f + \mathbf{MEM}^r_x F_{zr} - \mathbf{MEM}^f_y F_{zf} \sin\delta_f - C_A x_1^2}{m} + x_2 x_3, \\
f_2 &= \frac{\mathbf{MEM}^f_y F_{zf} \cos\delta_f + \mathbf{MEM}^r_y F_{zr} + \mathbf{MEM}^f_x F_{zf} \sin\delta_f}{m} - x_1 x_3, \\
f_3 &= \frac{a_1 \mathbf{MEM}^f_y F_{zf} \cos\delta_f + a_1 \mathbf{MEM}^f_x F_{zf} \sin\delta_f - a_2 \mathbf{MEM}^r_y F_{zr}}{I_z}.
\end{aligned}
\tag{14}
$$

The term $\dot{x}_1$ enters the right-hand side through (3), which makes the system an implicit ODE. In a rear-wheel-drive vehicle the front wheels roll freely, so $s_f = 0$ and, neglecting rolling resistance, $F_{xf} = 0$ [11]. This agrees with the finding that the main drift equilibria of RWD vehicles occur at near-zero front slip [5]. Equation (13) then reduces to

$$
\dot{x}_i = f_i(x_1, x_2, x_3, \dot{x}_1, s_r, \delta_f), \quad i = 1, 2, 3, \tag{15}
$$

which cannot be integrated directly by an explicit solver.

3 State-dependent coefficient (SDC) representation

To resolve the implicit structure and include first-order actuator dynamics, the state is augmented with the accelerations and the control signals,

$$
\boldsymbol{z} = [x_1,\ x_2,\ x_3,\ \dot{x}_1,\ \dot{x}_2,\ \dot{x}_3,\ s_r,\ \delta_f]^\top \in \mathbb{R}^8, \qquad \boldsymbol{U} = [\dot{s}_r,\ \dot{\delta}_f]^\top \in \mathbb{R}^2, \tag{16}
$$

and the control objective is to drive $\boldsymbol{z}$ to the target equilibrium $\boldsymbol{z}_{\text{eq}}$. Substituting into (15) gives $\dot{z}_i = f_i(z_1, z_2, z_3, z_4, z_7, z_8) \equiv f_i(\boldsymbol{z})$, $i = 1, 2, 3$. Differentiating in time, using $\dot{z}_{1,2,3} = z_{4,5,6}$, $\dot{z}_7 = U_1$, $\dot{z}_8 = U_2$, and solving the linear relation for $\dot{z}_4$ (which appears on both sides through f_{14}) yields

$$
\begin{aligned}
\dot{z}_{1,2,3} &= z_{4,5,6}, \qquad \dot{z}_7 = U_1, \quad \dot{z}_8 = U_2, \\
\dot{z}_4 &= \tfrac{1}{d}\big(f_{11} z_4 + f_{12} z_5 + f_{13} z_6 + f_{17} U_1 + f_{18} U_2\big), \\
\dot{z}_{3+i} &= \textstyle\sum_{c=1}^{3} \big(f_{ic} + \frac{f_{i4} f_{1c}}{d}\big) z_{3+c} + \sum_{j=7,8} \big(f_{ij} + \frac{f_{i4} f_{1j}}{d}\big) U_{j-6}, \qquad i = 2, 3,
\end{aligned}
\tag{17}
$$

where $d = 1 - f_{14}$ and $f_{ij} = \partial f_i / \partial z_j$. This is the SDC form

$$
\dot{\boldsymbol{z}} = \mathbf{A}(\boldsymbol{z})\, \boldsymbol{z} + \mathbf{B}(\boldsymbol{z})\, \boldsymbol{U}, \tag{18}
$$

with

$$\mathbf{A}(\boldsymbol{z}) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{f_{11}}{d} & \dfrac{f_{12}}{d} & \dfrac{f_{13}}{d} & 0 & 0 \\ 0 & 0 & 0 & f_{21} + \dfrac{f_{11} f_{24}}{d} & f_{22} + \dfrac{f_{12} f_{24}}{d} & f_{23} + \dfrac{f_{13} f_{24}}{d} & 0 & 0 \\ 0 & 0 & 0 & f_{31} + \dfrac{f_{11} f_{34}}{d} & f_{32} + \dfrac{f_{12} f_{34}}{d} & f_{33} + \dfrac{f_{13} f_{34}}{d} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{19}$$

$$\mathbf{B}(\boldsymbol{z}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \dfrac{f_{17}}{d} & \dfrac{f_{18}}{d} \\ f_{27} + \dfrac{f_{24} f_{17}}{d} & f_{28} + \dfrac{f_{24} f_{18}}{d} \\ f_{37} + \dfrac{f_{34} f_{17}}{d} & f_{38} + \dfrac{f_{34} f_{18}}{d} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{20}$$

All f_{ij} are evaluated by complex-step differentiation [21],

$$f_{ij} = \frac{\operatorname{Im}\bigl[f_i(\boldsymbol{z} + \mathrm{i}\, h\, \boldsymbol{e}_j)\bigr]}{h} + \mathcal{O}(h^2), \qquad h = 10^{-20}, \tag{21}$$

for $j \in \{1, 2, 3, 4, 7, 8\}$ only, since f_i does not depend on z_5 or z_6. The formula involves no subtraction of nearby values, so no step size has to be tuned against the narrow transition bands of the MEM switching functions. It requires f_i to be analytic, which holds when the absolute value and maximum are written in analytic form. All controllers use this evaluation.

Remark 1 (SDC factorisation compared with Taylor linearisation) The factorisation (18) is exact: the right-hand side of (17) equals $\mathbf{A}(\boldsymbol{z})\boldsymbol{z} + \mathbf{B}(\boldsymbol{z})\boldsymbol{U}$ for every $\boldsymbol{z}, \boldsymbol{U}$, so no affine term is needed. A first-order Taylor expansion of $\boldsymbol{g}(\boldsymbol{z}, \boldsymbol{U}) = \mathbf{A}(\boldsymbol{z})\boldsymbol{z} + \mathbf{B}(\boldsymbol{z})\boldsymbol{U}$ about $(\boldsymbol{z}_0, \boldsymbol{U}_0)$ has the Jacobian $\mathbf{A}_\mathrm{J} = \partial \boldsymbol{g}/\partial \boldsymbol{z}$, which requires the *second* derivatives of the tyre map, and an affine term. If the SDC matrices are frozen at $\hat{\boldsymbol{z}}$, the residual $[\mathbf{A}(\boldsymbol{z}) - \mathbf{A}(\hat{\boldsymbol{z}})]\boldsymbol{z}$ is $\mathcal{O}(\|\boldsymbol{z} - \hat{\boldsymbol{z}}\|)$, which is formally of lower order than the Taylor residual near $\hat{\boldsymbol{z}}$. The advantage of the SDC form is therefore not a smaller local error. First, the reference point can be moved along the predicted trajectory using first derivatives only, which keeps $\|\boldsymbol{z} - \hat{\boldsymbol{z}}\|$ small over the horizon. Second, the equilibrium is represented without an affine term (Proposition 1).

3.1 Embedded steady-state condition and structure of the SDC model

In the augmented model z_4, z_5, z_6 are *states*, which resolves the implicit ODE without an inner algebraic solver and places the steady-state condition inside the equations. Define the consistency residual

$$\boldsymbol{\varepsilon}(\boldsymbol{z}) = \begin{bmatrix} z_4 - f_1(\boldsymbol{z}) \\ z_5 - f_2(\boldsymbol{z}) \\ z_6 - f_3(\boldsymbol{z}) \end{bmatrix} \in \mathbb{R}^3, \tag{22}$$

which vanishes when (z_4, z_5, z_6) solves the implicit equation. The f_i are taken to be C^2 on the open region $\{d \neq 0\}$ where the model is used; in the implementation $|d| \geq 10^{-3}$.

Proposition 1 (Consistency manifold and embedded steady state) *Let $d(\boldsymbol{z}) \neq 0$ on the region considered.*

(a) Invariance. *Along every solution of (18) in this region, for any input $\boldsymbol{U}(t)$, $\mathrm{d}\boldsymbol{\varepsilon}/\mathrm{d}t = \mathbf{0}$. Hence $\mathcal{M} = \{\boldsymbol{z} : \boldsymbol{\varepsilon}(\boldsymbol{z}) = \mathbf{0}\}$ is invariant and locally a five-dimensional manifold, the graph of a map $(z_4, z_5, z_6) = \boldsymbol{\varphi}(z_1, z_2, z_3, z_7, z_8)$, on which the accelerations are the locally unique solution of the implicit equation.*

(b) Steady state. *A point $\boldsymbol{z}^* \in \mathcal{M}$ with $z^*_{4,5,6} = 0$ satisfies $f_i(z_1^*, z_2^*, z_3^*, 0, z_7^*, z_8^*) = 0$, which is the steady-state force and moment balance, and is an equilibrium of (18) for $\boldsymbol{U} = \mathbf{0}$. The target $\boldsymbol{z}_{\mathrm{eq}}$ of (23) is such a point.*

(c) Shift invariance. $\mathbf{A}(\boldsymbol{z})\boldsymbol{z}_{\mathrm{eq}} = \mathbf{0}$ *for every $\boldsymbol{z}$, so $\dot{\boldsymbol{e}} = \mathbf{A}(\boldsymbol{z})\boldsymbol{e} + \mathbf{B}(\boldsymbol{z})\boldsymbol{U}$ for $\boldsymbol{e} = \boldsymbol{z} - \boldsymbol{z}_{\mathrm{eq}}$, with no affine term and no trim input.*

The proof is given in Appendix A. By part (a), the steady-state condition need not be imposed as a constraint: the initial accelerations are computed from the implicit equation, which places the start on $\mathcal{M}$, and the trajectory remains there. In discrete time, integration error and state saturation cause a small drift off $\mathcal{M}$, which the closed loop treats as a disturbance. Part (c) allows the predictive controllers of section 4 to be formulated directly on $\boldsymbol{e}$.

Lemma 1 (Structural rank deficiency of the SDC pair) *Let $\mathbf{N}(\boldsymbol{z}) = \partial\boldsymbol{\varepsilon}/\partial\boldsymbol{z} \in \mathbb{R}^{3\times 8}$. For every $\boldsymbol{z}$ with $d(\boldsymbol{z}) \neq 0$:*

(i) $\operatorname{rank}[\mathbf{A}\ \ \mathbf{B}] = 5 < 8$;

(ii) the rows of $\mathbf{N}$ are linearly independent, with $\mathbf{NA} = \mathbf{0}$ and $\mathbf{NB} = \mathbf{0}$, so they span the left null space of $[\mathbf{A}\ \ \mathbf{B}]$;

(iii) the Hautus matrix $[-\mathbf{A}\ \ \mathbf{B}]$ at $\lambda = 0$ has rank 5, so the pair $(\mathbf{A}, \mathbf{B})$ has at least three uncontrollable modes at $\lambda = 0$, with left eigenvectors $\nabla\varepsilon_i(\boldsymbol{z})$.

The proof is given in Appendix B. The three uncontrollable zero modes correspond to the three conserved consistency coordinates; their number equals the codimension

of $\mathcal{M}$ and their directions are the gradients of ε_i. They are not physical vehicle modes. The 8-dimensional model is an explicit but non-minimal representation of the five-dimensional dynamics on $\mathcal{M}$, and controllability has physical meaning only within $\mathcal{M}$. The lemma does not exclude further uncontrollable modes at particular operating points, so the claim is "at least three".

Consequences for control synthesis. (i) The frozen pair never satisfies the stabilisability assumption of the algebraic Riccati equation on the full augmented space, because the uncontrollable eigenvalues lie at $\lambda = 0$. This concerns the redundant representation and not the stabilisability of the vehicle on $\mathcal{M}$ (see Remark 2). (ii) The MPC formulations use the matrices only as a prediction model and need no stabilisability assumption. (iii) The lemma applies to the frozen SDC pair and not to a Jacobian linearisation, which does not conserve $\boldsymbol{\varepsilon}$ away from equilibrium (Appendix C).

4 Controller design

4.1 Initial-condition severity framework

A controller comparison is meaningful only if the test conditions are ordered by their difficulty for the plant, independently of any controller. A two-criterion severity score is therefore used to rank the initial conditions (ICs) at $t = 0$. The target equilibrium is

$$\boldsymbol{z}_{\mathrm{eq}} = [12.4834,\ 5.1944,\ -0.4027,\ 0,\ 0,\ 0,\ 0.2,\ 0.3]^{\top} \tag{23}$$

(m/s, m/s, rad/s, …, s_r, rad), with body sideslip $\beta_{\mathrm{eq}} = \arctan(x_2^{\mathrm{eq}}/x_1^{\mathrm{eq}}) \approx 22.59°$ [13].

Criterion C_1: excess body sideslip,

$$C_1 = |\beta| - \beta_{\mathrm{eq}}, \quad \beta = \arctan(z_2/z_1), \tag{24}$$

measures the heading–velocity misalignment, which is associated with loss of steerability and imminent spin [13]. Only $|\beta|$ enters, so C_1 is independent of the sideslip direction and of the yaw rate.

Criterion C_2: normalised phase-plane recovery distance,

$$C_2 = \sqrt{\left(\frac{z_2 - z_{2,\mathrm{eq}}}{|z_{2,\mathrm{eq}}|}\right)^2 + \left(\frac{z_3 - z_{3,\mathrm{eq}}}{|z_{3,\mathrm{eq}}|}\right)^2}, \tag{25}$$

combines the lateral-velocity and yaw-rate deviations on a commensurate scale, in the spirit of phase-plane stability criteria [9]. Unlike C_1, it responds to a sign reversal of z_2 or z_3.

Each criterion is mapped to a rank $r \in \{1, 2, 3, 4\}$ by the fixed thresholds of table 3, and the composite score is $S_{\mathrm{comp}} = \frac{1}{2}(r_{C_1} + r_{C_2}) \in [1, 4]$ (equal weights, since neither criterion dominates).

Four ICs are used (tables 4 and 5). IC1–IC3 have increasing lateral and decreasing longitudinal velocity along the high-sideslip corridor of the equilibrium. IC4 is the

Table 3 Rank thresholds of the severity criteria.

Rank r	1	2	3	4
C_1 (°)	< 5	$[5, 15)$	$[15, 45)$	≥ 45
C_2 (–)	< 0.5	$[0.5, 1.5)$	$[1.5, 3)$	≥ 3

opposite-direction near-spin case: z_2, z_3 and δ_f have the opposite sign to the equilibrium, so the vehicle must reverse its yaw rate ($\Delta z_3 = -0.703\,\text{rad/s}$, about nine times $|\Delta z_3|$ of IC3). The initial accelerations are the solution of the implicit equation (Proposition 1): $(-2.967, -0.325, -2.439)$ for IC3 and $(-4.357, 2.631, 2.579)$ for IC4.

Table 4 Initial-condition state parameters. IC1–IC3 traverse the high-sideslip corridor; IC4 is the opposite-direction near-spin case requiring yaw-rate reversal.

IC	**Physical regime**	z_1 (m/s)	z_2 (m/s)	z_3 (rad/s)	δ_f (rad)	s_r (–)
Equilibrium	Reference state	12.48	+5.19	−0.40	+0.30	0.20
IC1	Near-nominal, slight drift	12.80	+5.50	−0.38	+0.28	0.20
IC2	Moderate sideslip	11.50	+7.00	−0.42	+0.32	0.25
IC3	Equal long./lat. speeds	9.50	+9.50	−0.48	+0.38	0.30
IC4	Near-spin, opposite direction	5.00	−20.00	+0.30	−0.42	0.52

Table 5 Two-criterion IC severity analysis at $t = 0$ relative to $\boldsymbol{z}_{\text{eq}}$. Ranks (r_{C_1}, r_{C_2}) follow table 3.

IC	$\lvert\beta\rvert$ (°)	C_1 (°)	C_2	(r_{C_1}, r_{C_2})	S_{comp}
Eq.	22.59	—	0.000	—	—
IC1	23.25	0.66	0.082	(1, 1)	1.00
IC2	31.33	8.74	0.350	(2, 1)	1.50
IC3	45.00	22.41	0.851	(3, 2)	2.50
IC4	75.96	53.37	5.155	(4, 4)	4.00

4.2 Normalised coordinates and common MPC formulation

SDRE and LMPC work in the physical state $\boldsymbol{z}$. LTV-MPC, NMPC and SDMPC work in normalised coordinates $\boldsymbol{\eta} = \mathbf{S}^{-1}\boldsymbol{z}$, $\mathbf{S} = \text{diag}(\boldsymbol{z}_{\text{eq}})$, with the zero entries of $\boldsymbol{z}_{\text{eq}}$ replaced by one, so that $\boldsymbol{\eta}_{\text{eq}} = (1, 1, 1, 0, 0, 0, 1, 1)^\top$. The dynamics become

$$\dot{\boldsymbol{\eta}} = \tilde{\mathbf{A}}(\boldsymbol{\eta})\boldsymbol{\eta} + \tilde{\mathbf{B}}(\boldsymbol{\eta})\boldsymbol{U}, \qquad \tilde{\mathbf{A}} = \mathbf{S}^{-1}\mathbf{A}(\mathbf{S}\boldsymbol{\eta})\mathbf{S}, \quad \tilde{\mathbf{B}} = \mathbf{S}^{-1}\mathbf{B}(\mathbf{S}\boldsymbol{\eta}), \tag{26}$$

with $\boldsymbol{U}$ unscaled. Since $\mathbf{S}$ is diagonal, $\tilde{\mathbf{A}}$ keeps the zero columns of $\mathbf{A}$, so Proposition 1(c) still holds, $\tilde{\mathbf{A}}\boldsymbol{\eta}_{\text{eq}} = \mathbf{0}$, and $\boldsymbol{e} = \boldsymbol{\eta} - \boldsymbol{\eta}_{\text{eq}}$.

LTV-MPC, NMPC and SDMPC share one optimal-control problem and differ only in how the prediction model is generated and the problem solved. With horizons N_p and $N_c \le N_p$ (inputs held for $k \ge N_c$),

$$J = \sum_{k=1}^{N_p-1} \boldsymbol{e}_k^\top \mathbf{Q}\, \boldsymbol{e}_k + \boldsymbol{e}_{N_p}^\top \mathbf{S}_N\, \boldsymbol{e}_{N_p} + \sum_{j=0}^{N_c-1} \left(\boldsymbol{U}_j^\top \mathbf{R}\, \boldsymbol{U}_j + \rho\, \|\boldsymbol{U}_j\|^2 \right), \qquad \rho = 10^{-6}, \quad (27)$$

with $\mathbf{Q} = \mathrm{diag}(250, 20, 2.5, 0, 0, 0, 0, 0)$, $\mathbf{R} = \mathrm{diag}(16, 8)$ and $\mathbf{S}_N = \mathrm{diag}(200, 20, 2, 0, 0, 0, 0, 0)$, subject to

$$\Delta \boldsymbol{U}_j = \boldsymbol{U}_j - \boldsymbol{U}_{j-1} \in [-(15,\, 1.5),\, (15,\, 1.5)]^\top, \qquad z_1 \ge 0.1, \;\; |s_r| \le 1, \;\; |\delta_f| \le 0.9, \quad (28)$$

where $\boldsymbol{U}_{-1}$ is the previously applied input and the state bounds act on all predicted states (mapped to $\boldsymbol{\eta}$ through $\mathbf{S}$, with bounds swapped for $z_{3,\mathrm{eq}} < 0$). Only the change of $\boldsymbol{U}$ per sample is limited. The plant clips the states to the same bounds. None of the predictive controllers has a terminal constraint or set, and $\mathbf{S}_N$ is a tuned weight, so no closed-loop stability or recursive-feasibility property is claimed for LMPC, LTV-MPC, SDMPC or NMPC.

4.3 State-dependent Riccati equation (SDRE) controller

SDRE treats the SDC pair as linear time-invariant at every instant and solves, for the frozen pair, the algebraic Riccati equation

$$\mathbf{A}^\top(\boldsymbol{z})\mathbf{P} + \mathbf{P}\mathbf{A}(\boldsymbol{z}) - \mathbf{P}\mathbf{B}(\boldsymbol{z})\mathbf{R}^{-1}\mathbf{B}^\top(\boldsymbol{z})\mathbf{P} + \mathbf{Q} = 0, \quad (29)$$

giving $\mathbf{K}(\boldsymbol{z}) = \mathbf{R}^{-1}\mathbf{B}^\top\mathbf{P}$ and the feedback $\boldsymbol{U} = -\mathbf{K}(\boldsymbol{z})(\boldsymbol{z} - \boldsymbol{z}_{\mathrm{eq}})$, which needs no feedforward term by Proposition 1(c). No prediction or constraint enters the synthesis. The CARE is solved with the MATLAB `lqr` routine in the physical state, with $\mathbf{Q} = \mathrm{diag}(10^3, 10^3, 10^3, 200, 200, 200, 10^{-3}, 10^{-3})$ and $\mathbf{R} = \mathbf{I}_2$, once per RK4 step at $\Delta t = 0.001\,\mathrm{s}$. If `lqr` fails or returns $\|\mathbf{K}\|_F \ge 10^8$, the last valid gain is reused, so the SDRE results describe this guarded implementation. There is no explicit input constraint, and states are clipped after each step.

Remark 2 (SDRE stability conditions) The SDRE law is locally asymptotically stabilising if the frozen pair is pointwise stabilisable and $(\mathbf{A}, \mathbf{Q}^{1/2})$ is pointwise detectable along the trajectory [14]. For the augmented model the first condition holds only in a degenerate sense. By Lemma 1, three modes at $\lambda = 0$ are uncontrollable and, for any gain $\mathbf{K}$, $\mathbf{A}-\mathbf{B}\mathbf{K}$ keeps three eigenvalues at zero (Appendix B). The CARE solution therefore gives no asymptotic-stability certificate in the 8-dimensional representation. On $\mathcal{M}$ the amplitudes of these directions are $\mathcal{O}(\|\boldsymbol{e}\|^2)$, consistent with the small residual error near equilibrium, but the gain remains ill-conditioned. In addition, $d = 1 - f_{14}$ approaches zero under strong load-transfer coupling and inflates the entries of $\mathbf{A}$ and $\mathbf{B}$. Both effects are consistent with the intermittent error spikes and the divergence from IC2 onward reported in section 5.2; their individual contributions were not separated. They are properties of this implicit-ODE factorisation and not a general limitation of SDRE.

4.4 Linear model predictive control (LMPC)

LMPC replaces the infinite-horizon solution with a receding-horizon QP [15] based on a single LTI model obtained by linearising the SDC model once at the equilibrium,

$$\mathbf{A}_d = \mathbf{I} + \Delta t\, \mathbf{A}(\boldsymbol{z}_{\mathrm{eq}}), \quad \mathbf{B}_d = \Delta t\, \mathbf{B}(\boldsymbol{z}_{\mathrm{eq}}), \quad \Delta t = 0.01\,\mathrm{s}. \tag{30}$$

With $\boldsymbol{e}_k = \boldsymbol{z}_k - \boldsymbol{z}_{\mathrm{eq}}$, $N_p = 20$ and $N_c = 15$, the lifted prediction is $\boldsymbol{E} = \boldsymbol{\Phi}\boldsymbol{e}_0 + \boldsymbol{\Gamma}\boldsymbol{U}$, with cost weights $\mathbf{Q}_L = \mathrm{diag}(10^3, 10^3, 10^3, 200, 200, 200, 0, 0)$ and $\mathbf{R}_L = \mathbf{I}_2$. The QP is

$$\min_{\boldsymbol{U}} \tfrac{1}{2}\boldsymbol{U}^\top \mathbf{H}\boldsymbol{U} + \boldsymbol{f}^\top \boldsymbol{U}, \qquad \mathbf{H} = \boldsymbol{\Gamma}^\top \bar{\mathbf{Q}}\boldsymbol{\Gamma} + \bar{\mathbf{R}} + 10^{-6}\mathbf{I}, \quad \boldsymbol{f} = \boldsymbol{\Gamma}^\top \bar{\mathbf{Q}}\boldsymbol{\Phi}\,\boldsymbol{e}_0, \tag{31}$$

subject to

$$\Delta \dot{s}_r \in [-3, 3], \quad \Delta \dot{\delta}_f \in [-0.3, 0.3]\ \mathrm{rad/s\ per\ sample}, \quad |s_r| \le 1, \quad |\delta_f| \le 0.9\,\mathrm{rad}, \quad z_1 \ge 0.1, \tag{32}$$

with state constraints acting on all predicted states. Since $\boldsymbol{\Phi}, \boldsymbol{\Gamma}, \mathbf{H}$ are constant and assembled offline, only $\boldsymbol{f}$ and the constraint right-hand sides change between samples. The QP is solved by `quadprog`; if it fails, the shifted previous solution is reused. LMPC differs from the other predictive controllers in more than the prediction model. It takes the input as zero after N_c, so s_r and δ_f stay constant there, whereas the others hold $\boldsymbol{U}_{N_c-1}$, and its horizon (0.2 against 0.6 s), discretisation and weights differ (table 8). Its results therefore cannot be attributed to the fixed model alone.

4.5 Linear time-varying MPC (LTV-MPC)

LTV-MPC re-linearises the nonlinear model at the current state at every sample and uses the resulting model over the whole horizon [15, 16]. It is the natural Jacobian-based benchmark for SDMPC. With $\boldsymbol{f}_\eta(\boldsymbol{\eta}, \boldsymbol{U}) = \tilde{\mathbf{A}}(\boldsymbol{\eta})\boldsymbol{\eta} + \tilde{\mathbf{B}}(\boldsymbol{\eta})\boldsymbol{U}$, the expansion about $(\boldsymbol{\eta}_i, \boldsymbol{U}_{i-1})$ is

$$\begin{aligned} &\dot{\boldsymbol{\eta}} \approx \mathbf{A}_{\mathrm{J}}\boldsymbol{\eta} + \mathbf{B}_{\mathrm{J}}\boldsymbol{U} + \boldsymbol{c}, \qquad \mathbf{A}_{\mathrm{J}} = \left.\frac{\partial \boldsymbol{f}_\eta}{\partial \boldsymbol{\eta}}\right|_{(\boldsymbol{\eta}_i, \boldsymbol{U}_{i-1})}, \quad \mathbf{B}_{\mathrm{J}} = \tilde{\mathbf{B}}(\boldsymbol{\eta}_i), \\ &\boldsymbol{c} = \boldsymbol{f}_\eta(\boldsymbol{\eta}_i, \boldsymbol{U}_{i-1}) - \mathbf{A}_{\mathrm{J}}\boldsymbol{\eta}_i - \mathbf{B}_{\mathrm{J}}\boldsymbol{U}_{i-1}. \end{aligned} \tag{33}$$

$\mathbf{B}_{\mathrm{J}}$ is exact because $\boldsymbol{f}_\eta$ is affine in $\boldsymbol{U}$. $\mathbf{A}_{\mathrm{J}}$ retains the term $\partial\tilde{\mathbf{A}}/\partial\boldsymbol{\eta}$, so second derivatives of the tyre map are required (Remark 1), and the affine term $\boldsymbol{c}$ is needed because the expansion point is not an equilibrium. Lemma 1 and Proposition 1(c) do not apply to this pair, and the Taylor model does not conserve $\boldsymbol{\varepsilon}$ exactly (Appendix C). The affine system is discretised exactly (zero-order hold) from one augmented exponential,

$$\exp\!\left(\Delta t \begin{bmatrix} \mathbf{A}_{\mathrm{J}} & \mathbf{B}_{\mathrm{J}} & \boldsymbol{c} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{A}_d & \mathbf{B}_d & \boldsymbol{c}_d \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \ \text{(top block rows)}, \tag{34}$$

giving $\boldsymbol{e}_{k+1} = \mathbf{A}_d\boldsymbol{e}_k + \mathbf{B}_d\boldsymbol{U}_k + \boldsymbol{w}$ with $\boldsymbol{w} = (\mathbf{A}_d - \mathbf{I})\boldsymbol{\eta}_{\mathrm{eq}} + \boldsymbol{c}_d$. The same $\mathbf{A}_d$ is used at every prediction step, so $\boldsymbol{\Phi}_i = \mathbf{A}_d^i$ and a constant drift $\boldsymbol{d}$ accumulates $\boldsymbol{w}$:

$\boldsymbol{E} = \boldsymbol{\Phi}\boldsymbol{e}_0 + \boldsymbol{\Gamma}\boldsymbol{U} + \boldsymbol{d}$. The QP is that of (27)–(28) with $\boldsymbol{f} = \boldsymbol{\Gamma}^\top \bar{\mathbf{Q}}(\boldsymbol{\Phi}\boldsymbol{e}_0 + \boldsymbol{d})$. Each sample requires one Jacobian, one 11×11 matrix exponential and the QP, which makes LTV-MPC the cheapest of the nonlinear-model controllers. Its model error, however, grows with the distance travelled by the state over the horizon.

4.6 Nonlinear MPC with CasADi and IPOPT (NMPC)

NMPC solves, at every sample, the nonlinear programme obtained from the nonlinear prediction model without freezing or linearisation. It is the accuracy reference of the comparison. It uses the same augmented dynamics as the plant, with the Jacobian $\partial\boldsymbol{f}/\partial\boldsymbol{z}$ obtained by algorithmic differentiation (CasADi `SX`) [19] and the accelerations from the same elimination as in (17), with $|d| \geq 10^{-3}$. One RK4 step gives $\boldsymbol{\eta}_{k+1} = \boldsymbol{F}_{\mathrm{RK4}}(\boldsymbol{\eta}_k, \boldsymbol{U}_k)$. Over 43 test states the relative mismatch to the plant step is 2.1×10^{-16}, so the NMPC model is identical to the plant. Direct multiple shooting is used,

$$\min_{\boldsymbol{\eta}_{0:N_p},\, \boldsymbol{U}_{0:N_c-1}} \; J \text{ of (27)} \quad \text{s.t.} \quad \boldsymbol{\eta}_0 = \boldsymbol{\eta}(t), \quad \boldsymbol{\eta}_{k+1} = \boldsymbol{F}_{\mathrm{RK4}}(\boldsymbol{\eta}_k, \boldsymbol{U}_{\min(k,N_c-1)}), \quad (28), \tag{35}$$

with 118 variables and 104 equality constraints for $N_p = 12$, $N_c = 7$. The problem is built with the CasADi `Opti` interface and solved by IPOPT [20] with an L-BFGS Hessian, tolerance 10^{-6} (acceptable 10^{-4}), at most 200 iterations and a warm start from the shifted previous solution. A solver failure (failure flag or exception) is handled by using the last iterate if finite, and otherwise the next input of the previous plan, limited by the increment bounds. Since cost, weights, horizons, constraints and plant are those of the other predictive controllers, any closed-loop difference comes from how the problem is posed and solved and not from model error. Tuned implementations (exact Hessians, real-time-iteration SQP [17, 18], compiled code) can be considerably faster, and the NMPC results describe the configuration of table 8.

4.7 State-dependent model predictive control (SDMPC)

Position among MPC schemes. LMPC freezes the model at one operating point, LTV-MPC at the current state, and NMPC keeps the full nonlinearity at the price of an NLP. SDMPC keeps the convex QP of linear MPC but builds the prediction model from the SDC factorisation evaluated along a nonlinear trajectory predicted over the horizon, needing only first derivatives of the tyre map. The distance between each evaluation point and the state to which the matrices are applied therefore stays small, which is where a single Taylor expansion degrades (Remark 1). SDMPC can be read as one frozen-coefficient (Picard) iteration of the NMPC problem, warm-started with the shifted previous solution, in the spirit of real-time-iteration schemes [17, 18]. The analogy has limits: those schemes use the exact sensitivity of the trajectory to the input plan, whereas the frozen SDC model omits the term $(\partial\tilde{\mathbf{A}}/\partial\boldsymbol{\eta})\boldsymbol{\eta}$. A fixed point of the scheme is therefore not in general a stationary point of the NMPC problem, and no convergence property is claimed.

Predicted trajectory and discretisation. At sample i the warm-start sequence $\hat{\boldsymbol{U}}$ (shifted previous solution, last input repeated) is applied to (26) by RK4 from $\boldsymbol{\eta}_i$,

giving $\{\hat{\boldsymbol{\eta}}_k\}_{k=0}^{N_p}$ clipped to the state bounds. The matrices $(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$ are evaluated at each $\hat{\boldsymbol{\eta}}_k$, $k = 0, \dots, N_p - 1$ (complex step, (21)), and discretised as

$$\mathbf{A}_{d,k} = \exp\big(\Delta t\, \tilde{\mathbf{A}}(\hat{\boldsymbol{\eta}}_k)\big), \qquad \mathbf{B}_{d,k} = \Delta t\, \tilde{\mathbf{B}}(\hat{\boldsymbol{\eta}}_k). \tag{36}$$

The input matrix is the first-order term of the zero-order-hold integral, because $\tilde{\mathbf{A}}^{-1}(\mathbf{A}_{d,k} - \mathbf{I})\tilde{\mathbf{B}}$ cannot be formed ($\tilde{\mathbf{A}}$ is always singular, Lemma 1). The neglected term $\frac{1}{2}\Delta t^2 \tilde{\mathbf{A}}\tilde{\mathbf{B}} + \dots$ is second order in $\Delta t = 0.05\,\mathrm{s}$ and could be recovered from the augmented exponential of (34) with $\boldsymbol{c} = \mathbf{0}$. Since LTV-MPC uses the exact ZOH, the discretisation cannot explain an SDMPC advantage.

Batch prediction model. By Proposition 1(c), $\boldsymbol{e}_{k+1} = \mathbf{A}_{d,k}\boldsymbol{e}_k + \mathbf{B}_{d,k}\boldsymbol{U}_k$ with no affine term, because the zero columns of $\tilde{\mathbf{A}}$ remain unit columns of $\mathbf{A}_{d,k}$, so $(\mathbf{A}_{d,k} - \mathbf{I})\boldsymbol{\eta}_{\mathrm{eq}} = \mathbf{0}$. Recursion gives, for $i = 1, \dots, N_p$,

$$\boldsymbol{e}_i = \Big(\prod_{k=i-1}^{0} \mathbf{A}_{d,k}\Big)\boldsymbol{e}_0 + \sum_{j=0}^{i-1}\Big(\prod_{k=i-1}^{j+1} \mathbf{A}_{d,k}\Big)\mathbf{B}_{d,j}\,\boldsymbol{U}_{\min(j,N_c-1)}, \tag{37}$$

with products ordered from the latest to the earliest factor since the $\mathbf{A}_{d,k}$ do not commute. Stacking,

$$\boldsymbol{E} = \boldsymbol{\Phi}\,\boldsymbol{e}_0 + \boldsymbol{\Gamma}\,\boldsymbol{U}, \qquad \boldsymbol{\Phi} = \begin{bmatrix} \mathbf{A}_{d,0} \\ \mathbf{A}_{d,1}\mathbf{A}_{d,0} \\ \vdots \\ \prod_{k=N_p-1}^{0} \mathbf{A}_{d,k} \end{bmatrix}, \qquad \boldsymbol{\Gamma}_{i,j} = \begin{cases} \Big(\prod_{k=i-1}^{j+1} \mathbf{A}_{d,k}\Big)\mathbf{B}_{d,j}, & i \geq j, \\ \mathbf{0}, & i < j, \end{cases} \tag{38}$$

where for $j = N_c - 1$ the block also collects the contributions $\mathbf{B}_{d,k}$, $k = N_c, \dots, N_p - 1$, of the held input. Both matrices are recomputed at every sample.

QP and cost. With $\bar{\mathbf{Q}} = \mathrm{blkdiag}(\mathbf{Q}, \dots, \mathbf{Q}, \mathbf{S}_N)$ and $\bar{\mathbf{R}} = \mathrm{blkdiag}(\mathbf{R}, \dots, \mathbf{R})$, the cost (27) becomes

$$\min_{\boldsymbol{U}} \ \tfrac{1}{2}\boldsymbol{U}^\top \mathbf{H}\boldsymbol{U} + \boldsymbol{f}^\top \boldsymbol{U}, \qquad \mathbf{H} = \boldsymbol{\Gamma}^\top \bar{\mathbf{Q}}\boldsymbol{\Gamma} + \bar{\mathbf{R}} + \rho\mathbf{I}, \quad \boldsymbol{f} = \boldsymbol{\Gamma}^\top \bar{\mathbf{Q}}\boldsymbol{\Phi}\,\boldsymbol{e}_0, \tag{39}$$

subject to (28), written as $\mathbf{A}_c\boldsymbol{U} \leq \boldsymbol{b}_c$ with constant increment rows and state rows $\pm\boldsymbol{\Gamma}\boldsymbol{U} \leq \boldsymbol{b}_x(\boldsymbol{e}_0)$. The QP has $n_u N_c = 14$ variables and, since $\mathbf{R} \succ 0$, a unique solution. If it is not solved or returns non-finite values, the previous plan is reused. Only $\boldsymbol{U}_0^*$ is applied (table 6). Per sample, SDMPC evaluates $(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$ at $4N_p$ points for the rollout and N_p points for the matrices (60 evaluations), computes N_p matrix exponentials of size 8×8 and solves one small QP. The workload is fixed, with no state-dependent iteration count.

Table 6 SDMPC algorithm executed at each sampling instant t. Steps marked ($\star$) replace the Jacobian-based linearisation of LTV-MPC; the rest follow the standard receding-horizon structure [15].

Step	Operation
1	**State measurement.** Obtain $\boldsymbol{\eta}_i = \mathbf{S}^{-1}\boldsymbol{z}(t)$ and $\boldsymbol{e}_0 = \boldsymbol{\eta}_i - \boldsymbol{\eta}_{\text{eq}}$.
2	**Warm start.** Shift the previous input sequence (repeating the last input) to obtain $\hat{\boldsymbol{U}}$.
3($\star$)	**Trajectory prediction.** Integrate (26) by RK4 from $\boldsymbol{\eta}_i$ with $\hat{\boldsymbol{U}}$ to obtain $\hat{\boldsymbol{\eta}}_{0:N_p}$.
4($\star$)	**SDC evaluation.** Evaluate $\tilde{\mathbf{A}}(\hat{\boldsymbol{\eta}}_k)$ and $\tilde{\mathbf{B}}(\hat{\boldsymbol{\eta}}_k)$, $k = 0, \ldots, N_p - 1$, via (21).
5($\star$)	**Discretisation and batch matrices.** Form $\mathbf{A}_{d,k}, \mathbf{B}_{d,k}$ by (36) and assemble $\boldsymbol{\Phi}, \boldsymbol{\Gamma}$ by (38); compute $\mathbf{H}, \boldsymbol{f}$.
6	**QP solve.** Solve (39) subject to (28) to obtain $\boldsymbol{U}^*$.
7	**Application.** Apply $\boldsymbol{U}_0^*$ for one sampling period and keep $\boldsymbol{U}^*$ for step 2 at $t + \Delta t$.

5 Simulation results

5.1 Simulation environment, controller settings and metrics

All simulations are deterministic and were run in MATLAB. The plant is the augmented nonlinear model (18), integrated by RK4 at the controller sampling time with the states clipped to their bounds. Tables 7 and 8 give the environment and the settings that determine each controller's behaviour. LTV-MPC, NMPC and SDMPC share cost, horizons, sampling time, constraints and plant. SDRE and LMPC appear only in section 5.2 and keep their design settings.

Metrics. The error norm is $\|\boldsymbol{e}(t)\|_2 = \|\boldsymbol{\eta}_{1:6}(t) - \boldsymbol{\eta}_{\text{eq},1:6}\|_2$ over the six normalised physical states; the actuator states are not weighted and are shown separately. The settling time is the last instant at which $\|\boldsymbol{e}\|_2 > 0.05$. The RMSE and ISE $= \int \|\boldsymbol{e}\|_2^2 \, \mathrm{d}t$ are taken over the whole run. The control effort $E = \sum_k \boldsymbol{U}_k^\top \mathbf{R} \boldsymbol{U}_k \, \Delta t$ uses the same expression for all controllers and serves only as a common yardstick. A condition is *stabilised* if the final error is finite and below 0.1.

5.2 SDRE versus LMPC

Both controllers use the SDC representation but handle nonlinearity and constraints differently (table 9). SDRE adapts the gain $\mathbf{K}(\boldsymbol{z})$ continuously but has no mechanism for explicit constraints, whereas LMPC uses a fixed linearisation and enforces hard bounds through the QP.

Near-nominal performance (IC1). Figure 2 compares the controllers from IC1 ($S_{\text{comp}} = 1.00$). SDRE converges in about 2 s, with aggressive early action that appears

Table 7 Numerical environment and timing protocol.

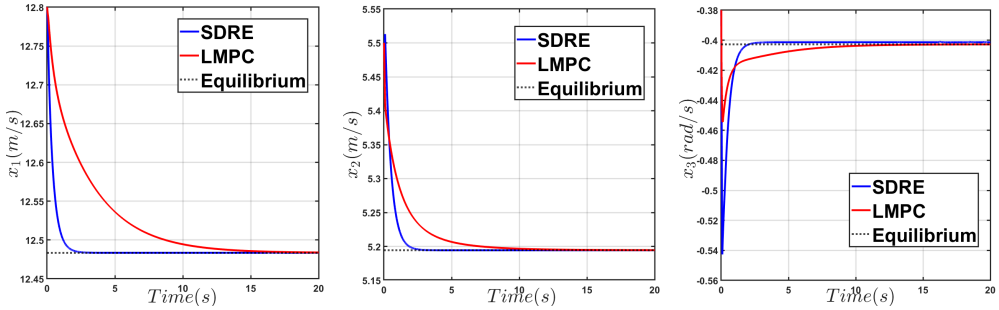
Item	Setting
Computing platform	
Processor	11th Gen Intel Core i7-1165G7, 2.80 GHz; 4 physical cores, 8 logical processors
Operating system	Windows, 64-bit
Software	MATLAB R2021b (9.11.0.1769968), 4 computational threads (default); Optimization Toolbox (`quadprog`, `fsolve`), Control System Toolbox (`lqr`)
NLP framework	CasADi 3.7.2 (MATLAB interface), `Opti` stack, bundled IPOPT
Numerical methods	
Plant integration	RK4 at the controller sampling time (0.001, 0.01 or 0.05 s, table 8); initial accelerations from `fsolve` (`TolFun` 10^{-10})
Jacobians of $\boldsymbol{f}$	Complex step, $h = 10^{-20}$; algorithmic differentiation inside NMPC
Protocol	
Test cases	ICs of table 4; 30 s runs (600 steps) for LTV-MPC, NMPC and SDMPC; 10–20 s for SDRE and LMPC
Timing	Wall-clock `tic/toc` per control step on the controller only (prediction, linearisation or matrix assembly, QP/NLP solve); plant integration excluded; one run per case

as large transients in $\dot{x}_1$ and $\dot{x}_2$, and a cumulative effort $J_{\text{ctrl}} = 5.7034$. LMPC converges smoothly by $t \approx 15$–20 s without overshoot, with $J_{\text{ctrl}} = 0.7079$ (88% lower) and hard bounds respected throughout. The phase portrait (figure 2g) shows a more direct but irregular SDRE path and a smoother curved LMPC arc. Both reach small final errors, but the SDRE error levels off near 3×10^{-3} with intermittent spikes at $t \approx 16$–19 s, which indicate occasional ill-conditioning of the Riccati gain (Remark 2).

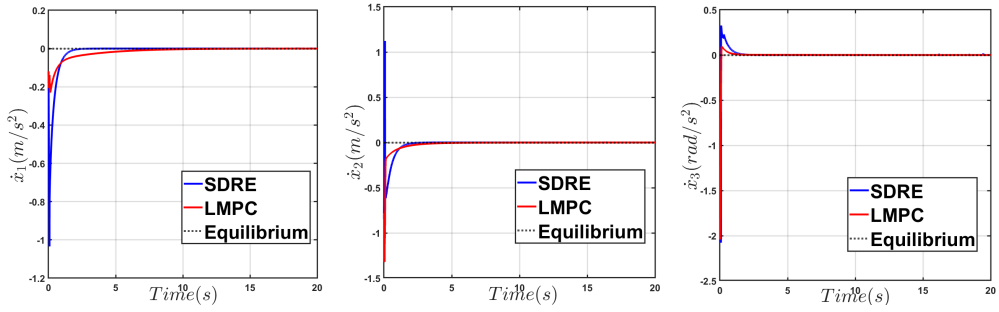
Reliability envelope. Neither architecture is reliable over the whole severity range (figure 3). SDRE fails from IC2 ($S_{\text{comp}} = 1.50$), where $d = 1 - f_{14}$ and the rank deficiency of the frozen pair make the gain ill-conditioned (Remark 2). LMPC stabilises IC2 but fails from IC3 ($S_{\text{comp}} = 2.50$), where the fixed linearisation no longer predicts the nonlinear dynamics adequately. This attribution was not isolated, because horizon, weights and input convention also differ from the other predictive controllers (section 4.4). The results show the limit of linear prediction: a controller reliable from IC3 onward must model the nonlinear dynamics inside the prediction loop, which motivates LTV-MPC and SDMPC.

5.3 LTV-MPC versus SDMPC

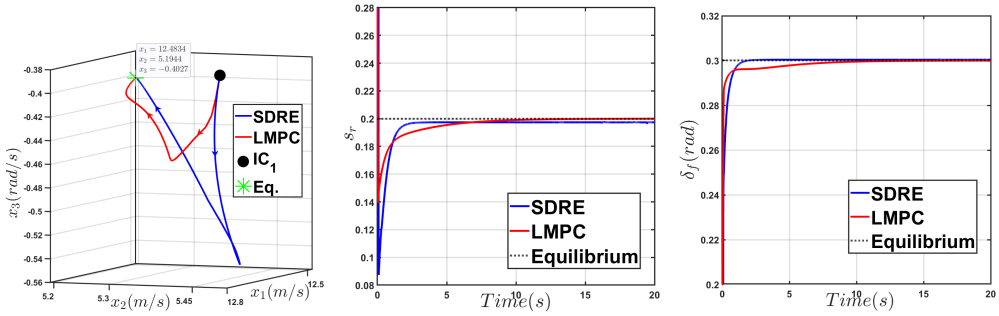
Both controllers place a state-varying model in the prediction loop and solve the same QP with the same weights, horizons ($N_p = 12$, $N_c = 7$, $\Delta t = 0.05$ s), constraints and plant. They differ in how the model is generated: one Taylor expansion at the current



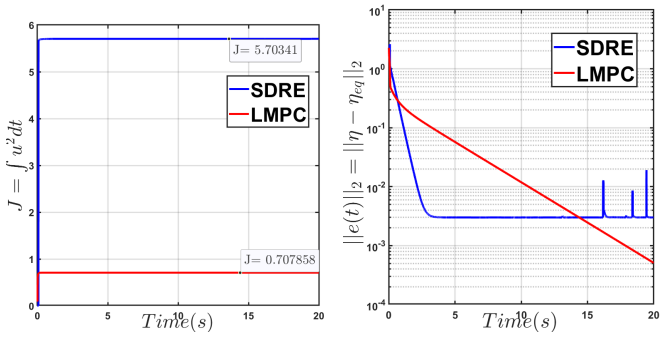
(a) Response of $x_1(z_1)$ (b) Response of $x_2(z_2)$ (c) Response of $x_3(z_3)$



(d) Response of $\dot{x}_1(z_4)$ (e) Response of $\dot{x}_2(z_5)$ (f) Response of $\dot{x}_3(z_6)$



(g) 3D phase portrait (h) Control input $s_r(z_7)$ (i) Control input $\delta_f(z_8)$



(j) Cumulative control effort (k) Normalised state error

Fig. 2 SDRE and LMPC responses from IC1 ($S_{\text{comp}} = 1.00$). SDRE converges in $\sim$2 s at 8$\times$ higher control effort; LMPC converges in $\sim$15–20 s with hard constraint satisfaction and 88% lower effort. Error spikes under SDRE near $t \approx 16$ s signal Riccati ill-conditioning.

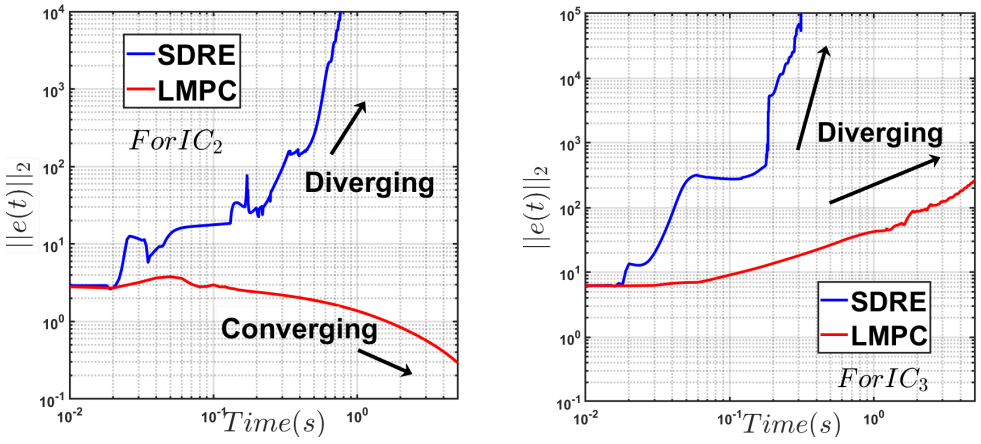
Table 8 Controller settings. $\Delta \boldsymbol{U}$: change of $\boldsymbol{U} = (\dot{s}_r, \dot{\delta}_f)$ between consecutive samples. All state bounds are $z_1 \geq 0.1\,\mathrm{m/s}$, $|s_r| \leq 1$, $|\delta_f| \leq 0.9\,\mathrm{rad}$.

	SDRE	**LMPC**	**LTV-MPC**	**SDMPC**	**NMPC**
State space	$\boldsymbol{z}$	$\boldsymbol{z}$	$\boldsymbol{\eta}$	$\boldsymbol{\eta}$	$\boldsymbol{\eta}$
Prediction model	CARE at current state	LTI at $\boldsymbol{z}_{\mathrm{eq}}$ (Euler)	Taylor at $(\boldsymbol{\eta}_i, \boldsymbol{U}_{i-1})$ + affine, exact ZOH	SDC along rolled-out trajectory; $\exp(\Delta t\tilde{\mathbf{A}})$, $\Delta t\tilde{\mathbf{B}}$	nonlinear, RK4 multiple shooting
Sampling time Δt	0.001 s	0.01 s	0.05 s	0.05 s	0.05 s
N_p/N_c (horizon)	–	20/15 (0.2 s)	12/7 (0.6 s)	12/7 (0.6 s)	12/7 (0.6 s)
State weight	10^3 ($z_{1–3}$), 200 ($z_{4–6}$), 10^{-3} ($z_{7,8}$)	10^3 ($z_{1–3}$), 200 ($z_{4–6}$), 0 ($z_{7,8}$)	$\mathbf{Q} = \mathrm{diag}(250, 20, 2.5, 0, 0, 0, 0, 0)$		
Input weight	$\mathbf{I}_2$	$\mathbf{I}_2$	$\mathbf{R} = \mathrm{diag}(16, 8)$, $\rho = 10^{-6}$		
Terminal weight	–	$= \mathbf{Q}$	$\mathbf{S}_N = \mathrm{diag}(200, 20, 2, 0, 0, 0, 0, 0)$		
Input limits	none	$\lvert\Delta \boldsymbol{U}\rvert \leq$ (3, 0.3)	$\lvert\Delta \boldsymbol{U}\rvert \leq$ (15, 1.5); $\boldsymbol{U}$ unbounded		
State limits	clipping only	in QP	in QP / NLP (all predicted states), and plant clipping		
Solver	`lqr`; last gain if it fails	`quadprog`	`quadprog`	`quadprog`	IPOPT, L-BFGS Hessian
Warm start	–	shifted plan	shifted plan	shifted plan (also seeds the rollout)	shifted primal plan
Input after N_c	–	zero	held at $\boldsymbol{U}_{N_c-1}$		
$(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$ evaluations per step	1	0 (offline)	1 + Jacobian	60	AD (symbolic)

Table 9 Algorithmic comparison of the SDRE and LMPC controllers.

Characteristic	**SDRE**	**LMPC**
Linearisation strategy	Online, state-dependent	Fixed at equilibrium
Control update rate	1 kHz ($\Delta t = 0.001\,\mathrm{s}$)	100 Hz ($\Delta t = 0.01\,\mathrm{s}$)
Optimisation	Analytical (CARE)	Numerical (QP)
Constraint handling	Implicit (penalty weights)	Explicit (hard constraints)
Gain computation	Adaptive $\mathbf{K}(\boldsymbol{z})$	Constant prediction matrices
Computational load	1000 CARE solves / s	100 QP solves / s

state (LTV-MPC) against SDC matrices along a rolled-out trajectory (SDMPC). This combines two differences, the number of linearisation points and the model type, which the experiments do not separate. IC3 ($S_{\mathrm{comp}} = 2.50$, where LMPC already fails) and IC4 ($S_{\mathrm{comp}} = 4.00$, requiring yaw-rate reversal) are analysed. The timings here come from an earlier derivative implementation and are comparable only between these two controllers (table 10, figure 4).



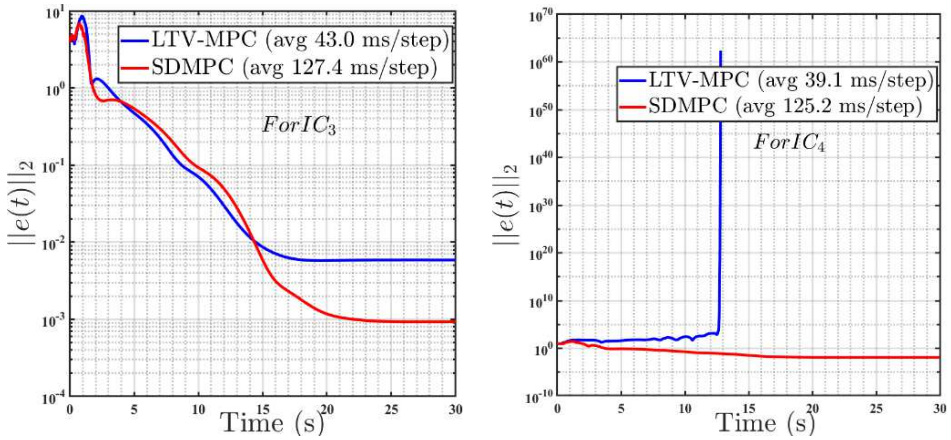
(a) Normalised error from IC2 (SDRE diverges; LMPC converges)

(b) Normalised error from IC3 (both controllers diverge)

Fig. 3 Normalised error norm under SDRE (blue) and LMPC (red) from progressively more severe initial conditions. SDRE fails from IC2 ($S_{\text{comp}} = 1.50$); LMPC fails from IC3 ($S_{\text{comp}} = 2.50$).

Table 10 LTV-MPC versus SDMPC on IC3 and IC4 (30 s, 600 steps, $\Delta t = 0.05$ s). Times are per control step for the controller only; the total is for 600 steps. The LTV-MPC run on IC4 diverges (error $> 10^{200}$ before $t = 15$ s, then overflow), so its RMSE and settling time are undefined and its timing is not representative.

IC	Method	Final $\|e\|$	RMSE	Settle (s)	Avg (ms)	Max (ms)	Total (s)	Effort E
IC3	LTV-MPC	0.005878	1.3918	10.95	43.03	105.83	25.82	18.09
IC3	SDMPC	0.000931	1.1538	12.00	127.41	237.05	76.44	14.36
IC4	LTV-MPC	Inf	Inf	–	39.07	343.20	23.44	1.49×10^{8}
IC4	SDMPC	0.012765	4.6804	14.20	125.16	204.35	75.10	67.97



(a) IC3 ($S_{\text{comp}} = 2.50$)

(b) IC4 ($S_{\text{comp}} = 4.00$)

Fig. 4 Error norm $\|e(t)\|_2$ (log scale) for LTV-MPC (blue) and SDMPC (red), with the average computation time per step. (a) Both converge, and SDMPC reaches a lower residual after $t \approx 14$ s. (b) LTV-MPC drifts, escapes near $t \approx 12.7$ s and diverges, whereas SDMPC converges.

IC3: both stabilise, with a speed–accuracy trade-off. LTV-MPC is 2.96× cheaper per step (43.0 against 127.4 ms) and settles about one second earlier (10.95 against 12.00 s). SDMPC overtakes it at $t \approx 14$ s: the LTV-MPC error stalls at 5.9×10^{-3} (cause not investigated), whereas SDMPC continues to 9.3×10^{-4}, a final error 6.3× smaller, with 1.21× lower RMSE and 21% lower effort (14.36 against 18.09). For a moderate transient the choice is a trade of computing time against steady-state accuracy and effort.

IC4: only SDMPC stabilises. LTV-MPC does not converge: its error is 40.8 at $t = 4.95$ s and 261 at 9.95 s, the system escapes near $t \approx 12.7$ s and the run overflows. SDMPC converges monotonically after the initial transient (0.811, 0.195 and 0.035 at 4.95, 9.95 and 14.95 s) to a final error of 0.0128, settling at 14.2 s with effort 68.0 at 125 ms per step. Along this path z_3 changes sign and z_2 travels 25 m/s, so the state leaves the neighbourhood of the expansion point within the 0.6 s horizon. We attribute the LTV-MPC failure to the resulting growth of the Taylor-model error, amplified by the second derivatives of the steeply switching ($\tanh(10^4 \cdot)$) tyre functions required by the Jacobian. SDMPC needs only first derivatives and re-evaluates the model along the predicted path. This mechanism was not isolated by ablation, and an LTV-MPC variant linearising along the predicted trajectory is not included.

5.4 SDMPC versus NMPC (CasADi/IPOPT)

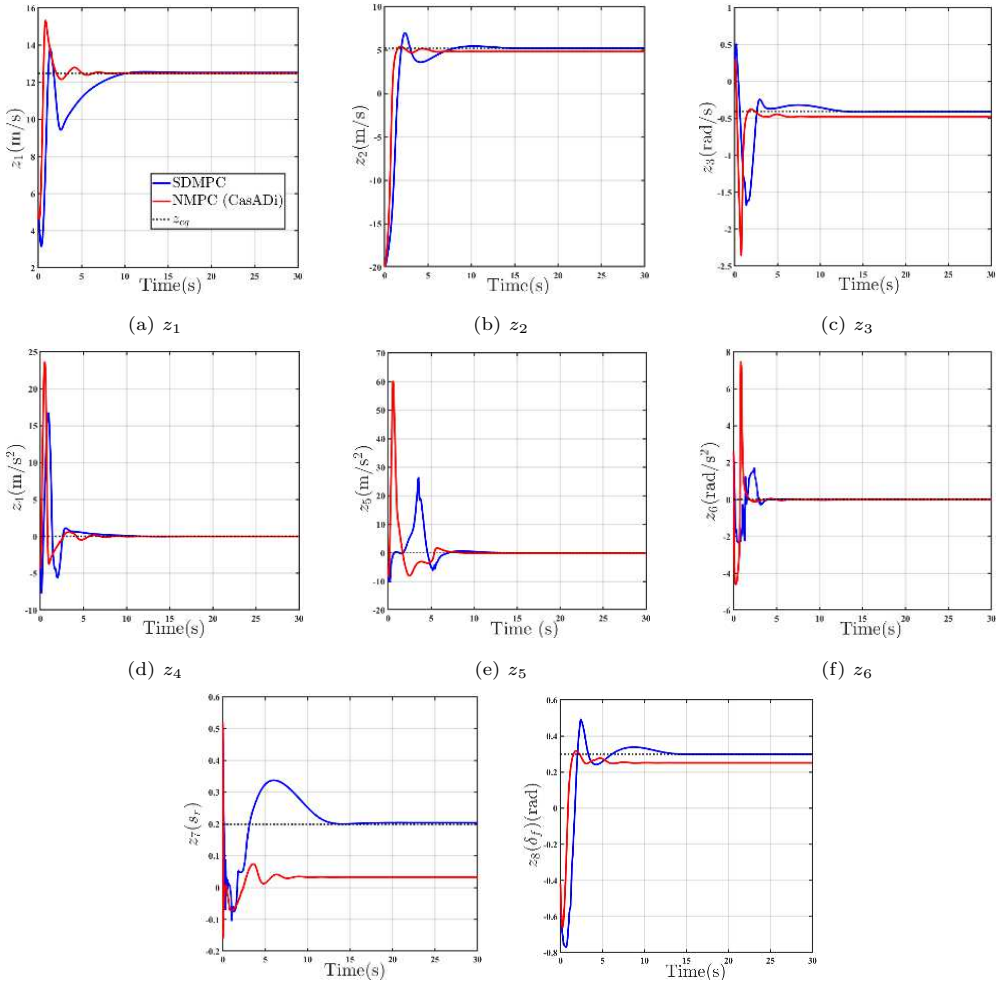
SDMPC is compared with NMPC on IC4. NMPC uses the plant model, cost and constraints of the other controllers, so the comparison isolates the effect of solving the nonlinear problem against one convex QP per step. The SDMPC timing here is lower than in section 5.3 because a different derivative implementation was used, and the two sets must not be mixed (table 11).

Table 11 SDMPC versus NMPC (CasADi/IPOPT) on IC4 (30 s, 600 steps, $\Delta t = 0.05$ s). ISE is $\int \|e\|_2^2 \, \mathrm{d}t$; "solver failures" counts the steps on which IPOPT reported failure (all in the first 5 s). The settling criterion is never met by NMPC.

Method	Final $\|e\|$	Settle (s)	ISE	Effort E	Avg (ms)	Max (ms)	Solver failures
SDMPC	0.013062	13.85	650.5	67.23	52.71	182.30	n/l[a]
NMPC	0.190879	never	929.7	160.31	600.39	1814.60	18 (3%)

[a]Not logged: the SDMPC script does not count QP fallbacks.

Accuracy. NMPC reduces the error faster initially (0.527 against 0.778 at $t = 5$ s) but stalls from $t \approx 10$ s at 0.1909, which is 14.6× the final SDMPC error of 0.0131 (itself still falling: 0.1745 at 10 s, 0.0301 at 15 s). The state plots show convergence to a *different steady state*: s_r and δ_f settle at about 0.03 and 0.25 rad instead of 0.2 and 0.3 rad, and z_2, z_3 at about 4.7 m/s and −0.48 rad/s (read from figure 5). SDMPC reaches all eight targets. Since the NMPC model coincides with the plant, the persistent offset is not caused by model error and points to the non-convex optimisation. Of the 600 solves, 18 (3%), all in the first 5 s, did not reach the requested tolerance. The



(a) z_1 (b) z_2 (c) z_3

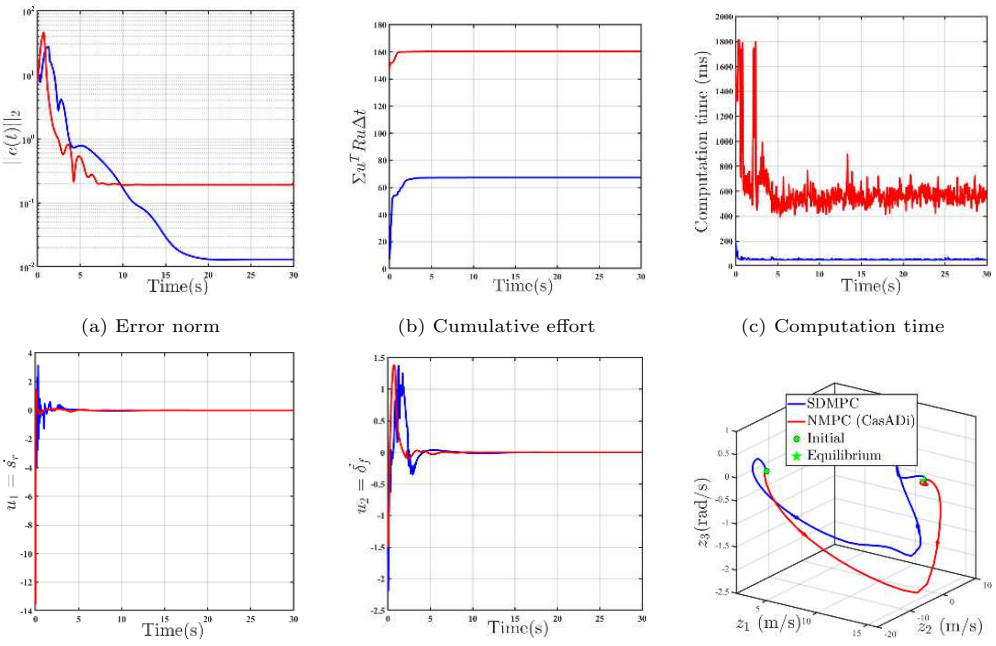
(d) z_4 (e) z_5 (f) z_6

(g) $z_7 = s_r$ (h) $z_8 = \delta_f$

Fig. 5 State responses from IC4 under SDMPC (blue) and NMPC (red); dotted lines show $\boldsymbol{z}_{\text{eq}}$. NMPC settles at a steady state that differs from the target in every panel, most visibly in z_7 and z_8.

local solutions found afterwards appear to hold the vehicle at a steady state that costs more than the target but from which the finite-horizon NLP finds no descent path within the iteration limit. This is an interpretation of the closed-loop behaviour and not a solver diagnosis, and it holds only for this configuration (quasi-Newton Hessian, 200 iterations).

Cost. NMPC effort is 2.4× higher (160.3 against 67.2) and its ISE 1.43× higher. Computation time is 600.4 against 52.7 ms on average (11.4×) and 1815 against 182 ms at the maximum (10×), with NMPC times above 1.5 s in places in the first seconds (figure 6c). Relative to the 50 ms sampling period, NMPC needs 12× real time and SDMPC 1.05×. Neither is real-time capable in this MATLAB prototype, but SDMPC,



(a) Error norm (b) Cumulative effort (c) Computation time

(d) $u_1 = \dot{s}_r$ (e) $u_2 = \dot{\delta}_f$ (f) Phase portrait (z_1, z_2, z_3)

Fig. 6 Performance of SDMPC (blue) and NMPC (red) from IC4: error norm, cumulative control effort $\sum \boldsymbol{U}^\top \mathbf{R}\boldsymbol{U}\,\Delta t$, computation time per step, control inputs and phase portrait.

with a fixed workload per step, is within reach of a compiled implementation, whereas NMPC is not.

5.5 Consolidated assessment

Table 12 consolidates the stabilisation outcomes, table 13 gives accuracy and cost of the nonlinear-prediction controllers on IC3 and IC4, and figure 7 shows the accuracy–time trade-off.

The results support four statements for this model, test set and implementation set. *Reliability:* SDMPC stabilised every condition with unchanged settings, whereas SDRE fails from IC2, LMPC from IC3, LTV-MPC diverges on IC4 and NMPC settles at an offset on IC4. *Accuracy against run time:* at IC3, LTV-MPC is 3× cheaper per step but ends with a 6× larger residual and 26% more effort; at IC4, LTV-MPC diverges and NMPC is 11× more expensive per step than SDMPC and less accurate in closed loop (0.19 against 0.013). *Predictable cost:* SDMPC performs a fixed amount of work per step (60 SDC evaluations, 12 matrix exponentials, one 14-variable QP) and its computation time is nearly constant (figure 6c), whereas the NMPC time varies by an order of magnitude with the iteration count. *Conditioning:* SDMPC shows no ill-conditioning of the kind seen with SDRE, with one tuning for all conditions and no gain scheduling.

Table 12 Consolidated stabilisation outcome across initial conditions of increasing severity. ✓: stabilised (final $\|\boldsymbol{e}\| < 0.1$); ×: failure; –: not simulated.

IC	Regime	S_{comp}	SDRE	LMPC	LTV-MPC	NMPC	SDMPC
IC1	Near-nominal	1.00	✓[a]	✓[b]	✓	✓	✓
IC2	Moderate slip	1.50	×[c]	✓	✓	✓	✓
IC3	Equal speeds	2.50	×	×[d]	✓[e]	✓	✓
IC4	Opposite near-spin	4.00	–[f]	–[f]	×[g]	×[h]	✓*

[a]Fast convergence, high effort. [b]Slow convergence, low effort. [c]Ill-conditioned Riccati gain. [d]Error grows. [e]Stalls at 5.9×10^{-3}. [f]Not simulated; the controller had already failed at a less severe condition. [g]Divergence. [h]Stalls at an offset steady state (0.19), 18 solver failures. *Only method that stabilises IC4.

Table 13 Accuracy and cost of the nonlinear-prediction controllers on IC3 and IC4. Run time is relative to the 50 ms sampling period. SDMPC is listed twice: (*a*) from the experiment of section 5.3, shared with LTV-MPC, and (*b*) from that of section 5.4, shared with NMPC; only ratios *within* an experiment are comparable.

IC	Method	Final $\|\boldsymbol{e}\|$	Avg time (ms)	Time/Δt	Effort E
IC3	LTV-MPC	0.0059	43.0	0.86	18.1
IC3	SDMPC[a]	0.00093	127.4	2.55	14.4
IC4	LTV-MPC	diverges	39.1	0.78	1.5×10^8
IC4	SDMPC[a]	0.0128	125.2	2.50	68.0
IC4	SDMPC[b]	0.0131	52.7	1.05	67.2
IC4	NMPC	0.191 (offset)	600.4	12.0	160.3

6 Conclusions

This paper sought a control method that balances accuracy and run time for a system implicit in the state derivative, using a rear-wheel-drive vehicle at the handling limit with longitudinal load transfer. Augmenting the state with the accelerations and actuator states resolves the implicit equation and gives an SDC model $\dot{\boldsymbol{z}} = \mathbf{A}(\boldsymbol{z})\boldsymbol{z} + \mathbf{B}(\boldsymbol{z})\boldsymbol{U}$ with an invariant consistency manifold of codimension three, an equilibrium without trim input and error dynamics without an affine term (Proposition 1). Five controllers were compared on one plant with initial conditions ordered by a two-criterion severity score (table 14).

- SDRE failed from IC2 and LMPC from IC3: frozen-gain and fixed-equilibrium models do not carry the nonlinearity of the implicit ODE away from equilibrium.
- LTV-MPC stabilised IC3 in a third of the SDMPC computing time, but with a six times larger final error and 26% more effort, and it diverged on IC4.

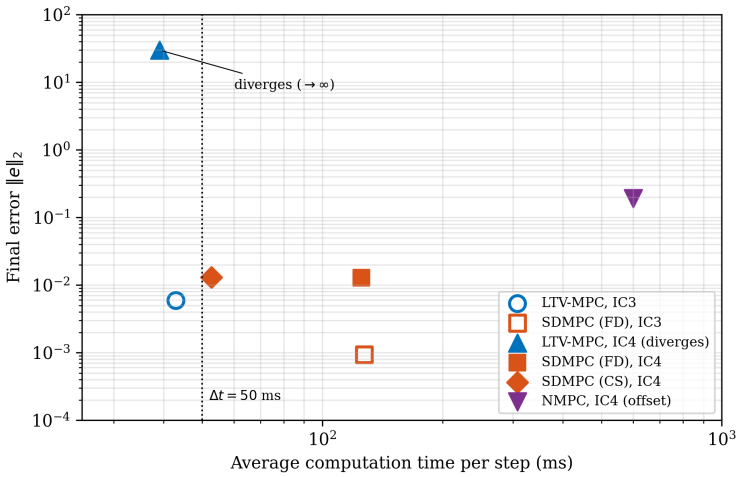


Fig. 7 Accuracy against average computation time per step. Filled markers: IC4; open markers: IC3. The diverged LTV-MPC run on IC4 is drawn at the upper edge. Moving down and to the left is better. SDMPC is the only controller that stabilises IC4, within about 1–2.5 sampling periods per step.

- NMPC, despite its exact model, settled at an offset steady state on IC4 (final error 0.191) after 18 solver failures, at 11× the SDMPC computing time and 2.4× its effort.
- SDMPC stabilised every condition with unchanged settings (final error 9.3×10^{-4} on IC3 and 1.3×10^{-2} on IC4) with a fixed workload per step.

SDMPC is therefore the most reliable controller within this study and, among those that stabilise IC4, the most accurate, at about one to two and a half sampling periods in a MATLAB prototype. It is not the cheapest (LTV-MPC is 3× faster on IC3), and its reliability claim rests on the severity range tested.

Limitations. (i) One vehicle model and four initial conditions were studied, and only IC4 discriminates between the predictive controllers. (ii) The LTV-MPC–SDMPC comparison changes the number of linearisation points and the model type together, and the LTV-MPC divergence mechanism was not isolated. (iii) NMPC uses one solver configuration and may improve with exact Hessians, real-time-iteration SQP or compiled code; its offset was not diagnosed beyond the failed solves. (iv) Timings are single-run MATLAB wall-clock values from different derivative implementations in sections 5.3 and 5.4, and neither SDMPC nor NMPC is real-time at $\Delta t = 0.05\,\mathrm{s}$ as implemented. (v) No stability or recursive-feasibility guarantee is given for any predictive controller, and a fixed point of the SDMPC iteration is not in general a stationary point of the NMPC problem. (vi) All simulations are noise-free with a perfect model and state measurement. (vii) LMPC differs from the other predictive controllers in several settings, so its failure is not attributed to the fixed model alone.

Future work. Natural next steps are a stability and feasibility analysis of SDMPC on the consistency manifold, an ablation against a trajectory-wise Jacobian LTV-MPC

and a real-time-iteration NMPC, compiled implementations with hardware-in-the-loop timing, robustness to model mismatch and sensor noise, and extension to more initial conditions and to trajectory tracking.

Table 14 Summary comparison of the five controllers on the implicit-ODE vehicle model. Run time is the average per step on IC4 (SDMPC from section 5.4, LTV-MPC from section 5.3; none for SDRE and LMPC, which ran at different sampling times). The highest severity stabilised is taken among IC1–IC4.

Method	**Prediction model**	**Per-step problem**	**Constraints**	**Highest IC stabilised**	**Run time (IC4)**
SDRE	Frozen SDC pair, no prediction	CARE	Not handled	IC1	n/a
LMPC	Linear, fixed at $\boldsymbol{z}_{\mathrm{eq}}$	Fixed-structure QP	Hard (QP)	IC2	n/a
LTV-MPC	Taylor at current state + affine	QP	Hard (QP)	IC3	39 ms (diverged)
NMPC	Exact nonlinear	Non-convex NLP (IPOPT)	Hard (NLP)	– (offset on IC4)	600 ms
SDMPC	Exact SDC along predicted trajectory	QP	Hard (QP)	**IC4**	53 ms

Declarations

Funding. This work was supported by the Fundamental Research Grant Scheme (FRGS) of the Ministry of Higher Education, Malaysia, under grant FRGS/1/2023/TK10/UPM/02/5.

Conflict of interest. The authors declare that they have no competing interests, financial or non-financial, that are directly or indirectly related to the work submitted for publication.

Data availability. The data that support the findings of this study are available from the corresponding author upon reasonable request.

Author contributions. J Zolfaghari: conceptualisation, methodology, software, formal analysis, investigation, writing – original draft. A S M Harithuddin: supervision, methodology, writing – review and editing. N A Aziz: supervision, writing – review and editing. R M K Raja Ahmad: writing – review and editing. R N Jazar: supervision, methodology, writing – review and editing.

Appendix A Proof of Proposition 1

A.1 Compact form. Let $\boldsymbol{z}_a = (z_4, z_5, z_6)^\top$. Since $f_{i5} = f_{i6} = 0$, define

$$\mathbf{F}_x = [f_{ic}]_{i,c=1}^{3}, \quad \mathbf{F}_u = [f_{ij}]_{i=1..3,\, j=7,8}, \quad \boldsymbol{\Delta} = \mathbf{I}_3 - \boldsymbol{f}_{\cdot 4}\boldsymbol{e}_1^\top = \begin{bmatrix} d & 0 & 0 \\ -f_{24} & 1 & 0 \\ -f_{34} & 0 & 1 \end{bmatrix}, \tag{A1}$$

with $\boldsymbol{f}_{\cdot 4} = (f_{14}, f_{24}, f_{34})^\top$ and $\det \boldsymbol{\Delta} = d$. Differentiating $\dot{z}_i = f_i(\boldsymbol{z})$ in time gives

$$\boldsymbol{\Delta}(\boldsymbol{z})\, \dot{\boldsymbol{z}}_a = \mathbf{F}_x\, \boldsymbol{z}_a + \mathbf{F}_u\, \boldsymbol{U}, \tag{A2}$$

and for $d \neq 0$, $\dot{\boldsymbol{z}}_a = \mathbf{M}_a \boldsymbol{z}_a + \mathbf{B}_a \boldsymbol{U}$ with $\mathbf{M}_a = \boldsymbol{\Delta}^{-1}\mathbf{F}_x$, $\mathbf{B}_a = \boldsymbol{\Delta}^{-1}\mathbf{F}_u$, whose entries are those in (19) and (20). With the ordering $(z_1, z_2, z_3 \,|\, z_4, z_5, z_6 \,|\, z_7, z_8)$,

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_a \\ \mathbf{I}_2 \end{bmatrix}. \tag{A3}$$

A.2 The identity $\mathbf{N}\boldsymbol{g} \equiv \mathbf{0}$. The Jacobian of $\boldsymbol{\varepsilon}$ is $\mathbf{N} = [-\mathbf{F}_x \;\; \boldsymbol{\Delta} \;\; -\mathbf{F}_u] \in \mathbb{R}^{3\times 8}$, since $\partial \varepsilon_i / \partial z_{3+k} = \delta_{ik} - f_{i4}\delta_{k1}$. For $\boldsymbol{g} = \mathbf{A}\boldsymbol{z} + \mathbf{B}\boldsymbol{U} = (\boldsymbol{z}_a,\ \mathbf{M}_a\boldsymbol{z}_a + \mathbf{B}_a\boldsymbol{U},\ \boldsymbol{U})$,

$$\mathbf{N}\boldsymbol{g} = (\boldsymbol{\Delta}\mathbf{M}_a - \mathbf{F}_x)\boldsymbol{z}_a + (\boldsymbol{\Delta}\mathbf{B}_a - \mathbf{F}_u)\boldsymbol{U} = \mathbf{0},$$

for every $\boldsymbol{z}$ with $d \neq 0$ and every $\boldsymbol{U}$, not only on $\mathcal{M}$.

A.3 Part (a). Along a solution in the region $d \neq 0$, $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\varepsilon}(\boldsymbol{z}(t)) = \mathbf{N}\boldsymbol{g} = \mathbf{0}$, so $\boldsymbol{\varepsilon}(\boldsymbol{z}(0)) = \mathbf{0}$ implies $\boldsymbol{\varepsilon}(\boldsymbol{z}(t)) = \mathbf{0}$ and $\mathcal{M}$ is invariant. Because $\partial\boldsymbol{\varepsilon}/\partial(z_4, z_5, z_6) = \boldsymbol{\Delta}$ has determinant $d \neq 0$, the implicit function theorem gives locally a unique C^1 map $\boldsymbol{\varphi}$ with $\boldsymbol{\varepsilon} = \mathbf{0} \iff \boldsymbol{z}_a = \boldsymbol{\varphi}(z_1, z_2, z_3, z_7, z_8)$, so $\mathcal{M}$ is locally a five-dimensional graph. The result is local: $\mathcal{M}$ may have several branches, so global uniqueness is not claimed, and the argument fails where $d = 0$.

A.4 Part (b). If $\boldsymbol{z}^* \in \mathcal{M}$ and $z^*_{4,5,6} = 0$, then $\varepsilon_i(\boldsymbol{z}^*) = 0$ reads $f_i(z_1^*, z_2^*, z_3^*, 0, z_7^*, z_8^*) = 0$. With $\boldsymbol{U} = \mathbf{0}$ and $\boldsymbol{z}_a = \mathbf{0}$, (A3) gives $\boldsymbol{g} = \mathbf{0}$, so $\boldsymbol{z}^*$ is an equilibrium. Conversely, a steady state with constant s_r, δ_f satisfies $f_i(x, 0, s_r, \delta_f) = 0$ and requires $\boldsymbol{U} = \mathbf{0}$ since $\dot{z}_7 = U_1$, $\dot{z}_8 = U_2$.

A.5 Part (c). By (A3), columns 1–3, 7, 8 of $\mathbf{A}$ vanish, and $\boldsymbol{z}_{\mathrm{eq}}$ has non-zero entries only in these positions, so $\mathbf{A}(\boldsymbol{z})\boldsymbol{z}_{\mathrm{eq}} = \mathbf{0}$. With $\boldsymbol{z} = \boldsymbol{e} + \boldsymbol{z}_{\mathrm{eq}}$ and $\dot{\boldsymbol{z}}_{\mathrm{eq}} = \mathbf{0}$, $\dot{\boldsymbol{e}} = \mathbf{A}\boldsymbol{e} + \mathbf{B}\boldsymbol{U}$. The absence of an affine term follows from the factorisation structure and the location of the equilibrium, not from linearisation. In normalised coordinates the same holds, $\tilde{\mathbf{A}}\boldsymbol{\eta}_{\mathrm{eq}} = \mathbf{0}$. □

Appendix B Proof of Lemma 1 and consequences

B.1 Rank (part i). By (A3), the non-zero columns of $[\mathbf{A}\ \ \mathbf{B}]$ form $\mathbf{C} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{M}_a & \mathbf{B}_a \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix} \in \mathbb{R}^{8\times 5}$. If $\mathbf{C}(\boldsymbol{a}, \boldsymbol{b})^\top = \mathbf{0}$, the first and last block rows give $\boldsymbol{a} = \mathbf{0}$, $\boldsymbol{b} = \mathbf{0}$, so the five columns are independent. The other five columns are zero, so $\operatorname{rank}[\mathbf{A}\ \ \mathbf{B}] = 5$.

B.2 Left null space (part ii). By (A2), $\mathbf{NA} = [\mathbf{0}\ \ -\mathbf{F}_x + \boldsymbol{\Delta}\mathbf{M}_a\ \ \mathbf{0}] = \mathbf{0}$ and $\mathbf{NB} = -\mathbf{F}_u + \boldsymbol{\Delta}\mathbf{B}_a = \mathbf{0}$. The rows of $\mathbf{N}$ are independent because its block $\boldsymbol{\Delta}$ is invertible. The left null space has dimension $8 - 5 = 3$, so it equals their span.

B.3 Hautus test (part iii). The Hautus matrix at $\lambda = 0$ is $[-\mathbf{A}\ \ \mathbf{B}]$, with rank 5 and corank three. Each $\boldsymbol{w}_i = \nabla\varepsilon_i$ satisfies $\boldsymbol{w}_i^\top \mathbf{A} = \mathbf{0}$ and $\boldsymbol{w}_i^\top \mathbf{B} = \mathbf{0}$, so it is a left eigenvector of $\mathbf{A}$ for $\lambda = 0$ annihilated by $\mathbf{B}$, which is the Hautus criterion for an uncontrollable mode. Equivalently, $\xi = \mathbf{N}\boldsymbol{e}$ obeys $\dot{\xi} = \mathbf{N}(\mathbf{A}\boldsymbol{e} + \mathbf{B}\boldsymbol{U}) = \mathbf{0}$ for the frozen pair. A larger number of uncontrollable modes at particular operating points is not excluded. □

B.4 Feedback invariance and size on $\mathcal{M}$. (1) For any $\mathbf{K} \in \mathbb{R}^{2\times 8}$, $\mathbf{N}(\mathbf{A} - \mathbf{BK}) = \mathbf{0}$, so $\mathbf{A} - \mathbf{BK}$ has three eigenvalues at 0, including for the SDRE gain, and the frozen closed loop cannot be asymptotically stable in the 8-dimensional space. (2) Let $\boldsymbol{z}, \boldsymbol{z}_{\text{eq}} \in \mathcal{M}$ with the segment between them in $\{d \neq 0\}$. Taylor's theorem gives $\boldsymbol{\varepsilon}(\boldsymbol{z}_{\text{eq}}) = \boldsymbol{\varepsilon}(\boldsymbol{z}) - \mathbf{N}(\boldsymbol{z})\boldsymbol{e} + \boldsymbol{r}$ with $\|\boldsymbol{r}\| \leq \frac{1}{2}\sup\|\nabla^2\boldsymbol{\varepsilon}\|\,\|\boldsymbol{e}\|^2$, hence $\mathbf{N}(\boldsymbol{z})\boldsymbol{e} = \mathcal{O}(\|\boldsymbol{e}\|^2)$. The uncontrollable coordinates are therefore second order in the error on $\mathcal{M}$, small near equilibrium but not identically zero.

Appendix C The Jacobian pair of LTV-MPC

Let $\mathbf{A}_{\text{J}} = \partial\boldsymbol{g}/\partial\boldsymbol{z}$ and $\mathbf{B}_{\text{J}} = \partial\boldsymbol{g}/\partial\boldsymbol{U} = \mathbf{B}(\boldsymbol{z})$ at $(\boldsymbol{z}, \boldsymbol{U})$ with $d \neq 0$.

Lemma 2 $\mathbf{NB}_{\text{J}} = \mathbf{0}$ *and* $\mathbf{A}_{\text{J}}^\top \nabla\varepsilon_i = -\nabla^2\varepsilon_i\,\boldsymbol{g}$, $i = 1, 2, 3$. *Hence* $\nabla\varepsilon_i$ *are left null vectors of* $\mathbf{A}_{\text{J}}$ *only where* $\boldsymbol{g} = \mathbf{0}$ *(equilibria) or* $\nabla^2\varepsilon_i\boldsymbol{g} = \mathbf{0}$.

Proof By A.2, $\nabla\varepsilon_i^\top \boldsymbol{g} = 0$ for all $\boldsymbol{z}$ with $d \neq 0$ and all $\boldsymbol{U}$. Differentiating with respect to $\boldsymbol{U}$ gives $\mathbf{NB} = \mathbf{0}$, and with respect to z_j gives $(\nabla^2\varepsilon_i\,\boldsymbol{g})_j + (\nabla\varepsilon_i^\top \mathbf{A}_{\text{J}})_j = 0$. □

Since $\nabla^2\varepsilon_i = -\nabla^2 f_i$ involves second derivatives of the tyre map, the Taylor model conserves $\boldsymbol{\varepsilon}$ only at equilibria, whereas the frozen SDC pair conserves it at every evaluation point (Lemma 1).

References

[1] Brenan, Kathryn Eleda, Stephen L. Campbell, and Linda Ruth Petzold. Numerical solution of initial-value problems in differential-algebraic equations. Society for Industrial and Applied Mathematics, 1995.

[2] Velenis, E., Frazzoli, E., Tsiotras, P.: On steady-state cornering equilibria for wheeled vehicles with drift. In: Proc. 48th IEEE Conf. Decision and Control, pp. 3545–3550 (2009)

[3] Velenis, E., Katzourakis, D., Frazzoli, E., Tsiotras, P., Happee, R.: Steady-state drifting stabilization of RWD vehicles. Control Eng. Pract. **19**(11), 1363–1376 (2011)

[4] Hindiyeh, R.Y., Gerdes, J.C.: A controller framework for autonomous drifting: design, stability, and experimental validation. J. Dyn. Syst. Meas. Control **136**(5), 051015 (2014)

[5] Milani, Sina, Hormoz Marzbani, and Reza N. Jazar. "Vehicle drifting dynamics: discovery of new equilibria." Vehicle system dynamics 60, no. 6 (2022): 1933-1958.

[6] Beal, C.E., Gerdes, J.C.: Model predictive control for vehicle stabilization at the limits of handling. IEEE Trans. Control Syst. Technol. **21**(4), 1258–1269 (2013)

[7] Wachter, E., Alirezaei, M., Bruzelius, F., Schmeitz, A.: Path control in limit handling and drifting conditions using state dependent Riccati equation technique. Proc. Inst. Mech. Eng. D J. Automob. Eng. **234**(2–3), 783–791 (2020)

[8] Goh, J.Y.M., Thompson, M., Dallas, J., Balachandran, A.: Beyond the stable handling limits: nonlinear model predictive control for highly transient autonomous drifting. Veh. Syst. Dyn. **62**(10), 2590–2613 (2024)

[9] Bobier-Tiu, C.G., Beal, C.E., Kegelman, J.C., Hindiyeh, R.Y., Gerdes, J.C.: Vehicle control synthesis using phase portraits of planar dynamics. Veh. Syst. Dyn. **57**(9), 1318–1337 (2019)

[10] Jazar, Reza N. Vehicle dynamics: theory and application. Springer Nature, 2025.

[11] Pacejka, H.B.: Tire and Vehicle Dynamics, 3rd edn. Butterworth-Heinemann, Oxford (2012)

[12] Zolfaghari, J., Harithuddin, A.S.M., Nuraini, A.A., Kamil, R., Jazar, R.N.: Modification of elliptical combined-slip tire formula in rear-wheel-drive vehicle. In: Dai, L., Jazar, R.N. (eds.) Nonlinear Approaches in Engineering Application. Springer, Cham (2026)

[13] Zolfaghari, Javad, A. S. M. Harithuddin, A. A. Nuraini, Raja Kamil, and Reza Jazar. "Nonlinear equilibrium analysis and codimension-one bifurcation of a three-state rear-wheel-drive vehicle model with combined-slip tire forces." Nonlinear Dynamics 114, no. 16 (2026): 1072.

[14] Cimen, T.: State-dependent Riccati equation (SDRE) control: a survey. IFAC Proc. Vol. **41**(2), 3761–3775 (2008)

[15] Rawlings, James B., David Q. Mayne, and Moritz M. Diehl. "Model predictive control: theory, computation, and design." (No Title) (2020).

[16] Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., Hrovat, D.: Predictive active steering control for autonomous vehicle systems. IEEE Trans. Control Syst. Technol. **15**(3), 566–580 (2007)

[17] Diehl, M., Bock, H.G., Schlöder, J.P.: A real-time iteration scheme for nonlinear optimization in optimal feedback control. SIAM J. Control Optim. **43**(5), 1714–1736 (2005)

[18] Gros, S., Zanon, M., Quirynen, R., Bemporad, A., Diehl, M.: From linear to nonlinear MPC: bridging the gap via the real-time iteration. Int. J. Control **93**(1), 62–80 (2020)

[19] Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M.: CasADi: a software framework for nonlinear optimization and optimal control. Math. Program. Comput. **11**(1), 1–36 (2019)

[20] Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)

[21] Martins, J.R.R.A., Sturdza, P., Alonso, J.J.: The complex-step derivative approximation. ACM Trans. Math. Softw. **29**(3), 245–262 (2003)